# Probability Hypothesis Densities for Multitarget, Multisensor Tracking with Application to Passive Radar

A Thesis
Presented to
The Academic Faculty

by

**Martin Tobias**

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Electrical and Computer Engineering
Georgia Institute of Technology
May 2006

# Probability Hypothesis Densities for Multitarget, Multisensor Tracking with Application to Passive Radar

Approved by:

Dr. Aaron D. Lanterman, Chair
School of Electrical and Computer Engineering
*Georgia Institute of Technology*

Dr. James H. McClellan
School of Electrical and Computer Engineering
*Georgia Institute of Technology*

Dr. Anthony J. Yezzi
School of Electrical and Computer Engineering
*Georgia Institute of Technology*

Dr. Douglas B. Williams
School of Electrical and Computer Engineering
*Georgia Institute of Technology*

Dr. Brani Vidakovic
School of Industrial and Systems Engineering
*Georgia Institute of Technology*

Date Approved: March 28, 2006

*This dissertation is dedicated to my parents, Raymond and Cynthia Tobias, and my grandparents, Issy and Freda (z"l) Kadish and Ralph (z"l) and Helene (z"l) Tobias.*

# ACKNOWLEDGEMENTS

My first "thank you" belongs to my advisor, Dr. Aaron Lanterman. Since our very first meeting, Aaron has displayed a genuine concern for both my professional and overall well-being, and I am very grateful. I thank him for ensuring that I was always funded, for connecting me with his professional contacts, and for providing continual encouragement and sage guidance in my research. Aaron is a source of continuous enthusiasm, from which I am immensely fortunate and delighted to have profited.

I sincerely thank Dr. Ronald Mahler of Lockheed Martin MS2 Tactical Systems for all the help he has provided in my work with the probability hypothesis density. To Dr. Paul Howland, I express a heartfelt "thank you" for providing me with an internship experience at the NATO C3 Agency in The Hague. While there, I also had the privilege of working with Darek Maksimiuk, whom I thank for familiarizing me with the NC3A passive radar system, and Kees Stolk, to whom I am grateful for many radar-related and tracking-related discussions.

Much thanks, too, is in order for the faculty and staff at the Center for Signal and Image Processing. In particular, I thank Christy Ellis, Erica Edwards, Renaldia Parker, and Carla Zachary for ensuring that everything ran smoothly for me at CSIP. I have also learned much and immensely enjoyed my time at CSIP due to the interaction with my fellow graduate students, especially those involved with the Student Action Committee. At the risk of forgetting someone, I will name a few of them: Amer Abufadel, David Boivin, Milind Borkar, Kevin Chan, Vince Emanuele, Michael Farrell, Majid Fozunbal, Woojay Jeon, Greg Krudysz, Clyde Lettsome, Sam Li, Maneli Noorkami, Soner Ozgur, Nazanin Rahnavard, Gail Rosen, Mina Sartipi, Nikolaos Vasiloglou, Jérôme Vasseur, and Rajbabu Velmurugan. Each one has positively affected my experience at Georgia Tech, and I am grateful for their acquaintance and friendship. I thank especially my labmate, Nicolas Gastaud, for making graduate school infinitely more enjoyable.

I sincerely thank Badri N. Vellambi R. for sharing some of his exceptional knowledge of mathematics with me. I especially thank him for his help with the mathematics in Sections 6.5.2 and 6.5.3 of this thesis. I also thank Volkan Cevher for imparting to me some of his vast knowledge and expertise on all things tracking and particle filtering. To Raviv Raich, I am grateful not only for his invaluable review of one of my papers, his LaTeX expertise, and his general advice, but also for his and Genevieve's friendship.

I express a particular thanks to my groupmates, Kerkil Choi, Jason Dixon, Lisa Ehrman, William Leven, Jonathan Morris, Ryan Palkki, and Cody Wortham for their support. From daily details to bigger issues, Will and Kerkil were constant sources of advice and always had an ear to lend as we progressed through graduate school together. I am very glad to have gotten to know Kerkil, and I feel fortunate, and deeply grateful, to have had Will as my "twin" at Georgia Tech.

To my roommate, Rungsun Munkong, I am enormously grateful for his patience and encouragement. Roy is both kind and considerate, and I thank him for his reassuring and enlightening words whenever I bothered him with my concerns.

I thank also all those friends who, both in Atlanta and from greater distances, have supported and encouraged me during the past few years. They know who they are, so I will not list them. Were I to do so, I could continue for many pages, as I am grateful to each and every one of them.

Lastly, I thank my brothers, Barry and Jeremy, and my parents, Raymond and Cynthia, for their continuous support. They were always a phone call, e-mail, or short drive away to lend an ear or a hand and to provide me with much advice, reason, and common sense. They also were, and still are, a source of much love and encouragement. For all this and more, I am extremely fortunate and profoundly grateful.

The work in this thesis was enabled via startup funds from the School of Electrical Engineering at the Georgia Institute of Technology, the Demetrius T. Paris Junior Professorship, U.S. Air Force Office of Scientific Research grant F49620-03-1-0340 and the support of Dr. Paul Howland and Dr. Rene van der Heiden of NATO C3 Agency.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# SUMMARY

The probability hypothesis density (PHD), popularized by Ronald Mahler, presents a novel and theoretically-rigorous approach to multitarget, multisensor tracking. Based on random set theory, the PHD is the first moment of a point process of a random track set, and it can be propagated by Bayesian prediction and observation equations to form a multitarget, multisensor tracking filter. The advantage of the PHD filter lies in its ability to estimate automatically the expected number of targets present, to fuse easily different kinds of data observations, and to locate targets without performing any explicit "report-to-track" association.

We apply a particle-filter implementation of the PHD filter to realistic multitarget, multisensor tracking using passive coherent location (PCL) systems that exploit "illuminators of opportunity" such as FM radio stations.

The objective of this dissertation is to enhance the usefulness of the PHD particle filter for multitarget, multisensor tracking, in general, and within the context of PCL, in particular. This involves a number of thrusts, including: 1) devising intelligent proposal densities for particle placement, 2) devising a peak-extraction algorithm for extracting information from the PHD, 3) incorporating realistic probabilities of detection and signal-to-noise ratios (including multipath effects) to model realistic PCL scenarios, 4) using range, Doppler, and direction of arrival (DOA) observations to test the target detection and data fusion capabilities of the PHD filter, and 5) clarifying the concepts behind FISST and the PHD to make them more accessible to the practicing engineer.

A goal of this dissertation is to serve as a tutorial for anyone interested in becoming familiar with the probability hypothesis density and associated PHD particle filter. It is hoped that, after reading this thesis, the reader will have gained a clearer understanding of the PHD and the functionality and effectiveness of the PHD particle filter.

# CHAPTER I

# INTRODUCTION

## *1.1 Review of Multitarget, Multisensor Tracking*

The theory behind single-target, single-sensor tracking is quite well understood. The primary tool used is the Kalman filter [33, 79], which works well when the target's motion and the sensor's observation model are both linear. Variations of the Kalman filter, such as the extended Kalman filter and newer unscented Kalman filter [67, 78], are often used when nonlinearities are present. Improved target tracking has also been obtained by enhancing the flexibility of the Kalman filters with algorithms such as the interacting multiple model (IMM) algorithm [7] or, more recently, by using adaptive neural networks [59]. A rather complete overview of classical multitarget tracking techniques is provided in [64]. Lately, with the ever increasing power of today's computing resources, excitement has arisen over the use of particle filters to facilitate target tracking in nonlinear and non-Gaussian scenarios [1, 8, 15, 23, 69, 76, 82]. Particle filters are discussed further in Chapter 3.

In the presence of multiple targets, tracking becomes more difficult, since one must determine which reports from a given sensor were generated by which targets. When multiple sensors are used, some form of data fusion is also needed to resolve the multiple reports created by a single target. Multitarget, multisensor tracking is, thus, a combination of these two complex tasks, the difficulty of which increases further in the presence of false alarms and missed detections. Furthermore, applying classical state estimators, such as the maximum *a posteriori* (MAP) or expected *a posteriori* (EAP), directly to the multitarget case leads to undefinable and arbitrary results (see Section 6.8 of [43]).

Two basic questions that must be addressed by the multitarget, multisensor tracker are:

1. How many targets are present?

2. How should it keep track of the targets?

Traditionally, multitarget tracking schemes have answered the latter question by using a method called "report-to-track" association. In such a technique, the multitarget tracker maintains a set of target tracks and attempts to fit the sensor measurements to these tracks. For example, in the joint probabilistic data association (JPDA) filter [2, 6], the sensor measurements are validated and assigned association probabilities, which are then used to update the set of target tracks at each scan. In multiple hypothesis tracking (MHT) [2, 6], sensor observations are used over multiple scans to determine possible target tracks, which are then used to extend existing tracks or add new tracks. In both methods, explicit associations are made between sensor observations and target tracks to determine the current target state.

Eliminating the report-to-track association may potentially avoid any introduction of error that may arise from incorrect associations, so alternative algorithms that do not require explicit report-to-track association have been proposed. One such technique is the symmetric measurement equations (SME) filter [34–36, 66]. It applies nonlinear symmetric functions to the measurements and uses the properties of symmetric equations to obtain target positions without doing any explicit report-to-track association. The work presented in this document is based on another associationless approach, one that relies on finite-set statistics (FISST).

FISST also nicely addresses the question as to how many targets are present, since it can be used to derive equations that explicitly estimate the expected number of targets present in the area under observation. This avoids the need to make any assumptions as to the number of targets present or to use ad-hoc methods to estimate the number of targets.

## *1.2 Finite-Set Statistics*

When an engineer thinks of random variables, useful terms from "engineering statistics" usually come to mind, such as mean, variance, probability density function, cumulative distribution function, moment, and characteristic function. To use these concepts, the target-tracking engineer does not need to remember the mathematical definition of a random variable, namely, that it is a function mapping a probability space to the space of real

numbers. Rather, "engineering statistics," which can be applied in a straightforward manner to random vectors, can be readily applied to the state vector in single-target tracking.

In an extensive series of conference papers (particularly the SPIE sessions on *Signal and Data Processing of Small Targets* and *Signal Processing, Sensor Fusion, and Target Recognition*), Mahler has proposed "random sets," which map elements of an underlying probability space to sets, as the most natural framework to address data fusion in general and target tracking systems in particular. Unfortunately, the probability theory associated with random sets is nowhere near as well-known as the theory associated with the more mundane random variables and vectors. However, the presentation of the theory in the book by Goodman, Mahler, and Nguyen [21] is thorough. The authors define "set integrals" and "set derivatives" in terms of generalized Radon-Nikodým derivatives. These set generalizations of calculus allow for generalizations of probability densities and distribution functions. Once the random sets are given a solid measure-theoretic foundation, ideas from statistics, such as maximum-likelihood estimation and maximum *a posteriori* estimation, and from information theory, such as entropy and Kullback-Leibler distances, can be extended to these random sets (see Sections 5.2 and 5.3 of [21]).

Mahler [43, 45] attempts to distill random set theory to nuts-and-bolts principles that practicing engineers can easily apply. One of the more elegant aspects of traditional Kalman filtering is the way in which the prior and posterior distributions are characterized by a small set of sufficient statistics, *viz.* a mean and covariance, that are easily propagated in the Kalman recursion. When target tracking is generalized to the multitarget, multisensor scenario, however, no simple analogous implementation seems to appear.

Nevertheless, attempting to replicate the simplicity of the Kalman filter for the multitarget, multisensor case, Mahler and Zajic [53] propose propagating the first moment of a function that maps a set of targets into a continuous function space. This functional mapping is essential since the expectation of a set-valued random variable is not well-defined. They choose a function that places Dirac deltas at the target positions and call its first moment a probability hypothesis density (PHD). The PHD acts much like an intensity of a Poisson point process; in fact, it is the first factorial moment density found in point process

theory [14, 46]. Like the mean and variance of the Kalman filter, the PHD is readily propagated forward through the Bayesian prediction and data-update steps. It is, of course, a bit more complicated since an entire function is being propagated forward, not just a mean vector and a covariance matrix.

According to Howland *et al.* [28], the biggest challenge in implementing a passive radar tracker that exploits multiple FM radio transmitters is developing a technique to fuse multiple observations of the same target together. The PHD filter automatically handles the non-trivial tasks of both target number estimation and data fusion. There have been a number of applications of a particle-filter implementation of the PHD filter to multitarget, multisensor scenarios [13, 68, 76, 77, 83], including one using real forward-scan sonar image data [12]. However, unlike the work presented in this thesis, none of these applications include both false alarms and realistically fluctuating probabilities of detection.

## *1.3 Thesis Organization*

FISST and the Bayesian PHD-based filtering equations are explained in further detail in Chapter 2. Chapter 3 contains an introduction to particle filtering and describes the particle filter implementation of the Bayesian PHD filter. Chapter 4 provides an account of our initial application of the PHD particle filter to a simulated, realistic passive-radar scenario, where we also compare the tracking performance of the filter when using range-only observations to when using both range and Doppler observations. A technique of placing particles along the edges of the field of view is introduced, and an expectation-maximization (EM) algorithm for finding the target states by extracting the peaks in the PHD is presented. A subsequent simulation, containing improvements in the functionality of the PHD particle filter implementation and increased realism in the passive radar scenario, is detailed in Chapter 5. The EM algorithm is replaced by a simpler, more effective peak-extraction algorithm, and a new technique that places particles throughout the scene at bistatic range ellipse and Doppler observation intersections is used. Further improvements in the realism of the simulations, such as incorporating multipath effects, and in the functionality of the

PHD-based tracker, such as adding direction of arrival observations, are presented in Chapter 6. Chapter 7 summarizes the contribution and conclusions of our research and discusses possible avenues of future work.

# CHAPTER II

# FINITE SET STATISTICS

## *2.1 Single Target, Single Sensor Tracking*

Before delving into multitarget, multisensor tracking, it is worthwhile to begin with the single target, single sensor tracking case. Consider the scenario in which a single target is present. We want to detect this target using a single sensor that provides some sort of observation $\mathbf{z}_k$ at each time step $k$. At any time $k$, then, we have a collection consisting of the current observation and all past observations. We denote this by the set $Z^k = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_k\}$. Given these observations, we want to find the state $\mathbf{x}$ of the target. In the Bayesian framework for detection, this is achieved by using the Bayesian posterior density $f_{k|k}(\mathbf{x}|Z^k)$, which is the likelihood that the target has state $\mathbf{x}$ at time $k$, given the set of observations up to time $k$. The state $\mathbf{x}$ of the target can contain information about the target location, target velocity, or any other identifying parameter, and it is estimated from the posterior density by taking either the maximum *a posteriori (MAP)* estimate,

$$\hat{x}_{k|k}^{MAP} = \arg\sup_x f_{k|k}(\mathbf{x}|Z^k), \tag{1}$$

or the expected *a posteriori (EAP)* estimate, also known as the conditional mean estimate (CME),

$$\hat{x}_{k|k}^{EAP} = \int \mathbf{x} \cdot f_{k|k}(\mathbf{x}|Z^k) d\mathbf{x}. \tag{2}$$

Having extracted the target state at time step $k$, we now use the single target Bayes filter to find the Bayesian posterior density at time $k+1$. This filtering is achieved in two steps. The first step is to apply the Time Update equation, which exploits our *a priori* knowledge of the target's motion to predict where the target will be at time $k+1$. The *a priori* knowledge of the target motion is embodied in $f_{k+1|k}(\mathbf{y}|\mathbf{x})$, which assumes a Markov model of the target motion and is the likelihood that a target with state $\mathbf{x}$ at time $k$ will have state $\mathbf{y}$ at time $k+1$. Using this target motion likelihood and the Bayesian posterior

density of time step $k$, a time-updated "prediction" density is obtained via the Time Update equation:

$$f_{k+1|k}(\mathbf{x}|Z^k) = \int f_{k+1|k}(\mathbf{x}|\mathbf{w}) f_{k|k}(\mathbf{w}|Z^k) d\mathbf{w}. \tag{3}$$

The second part of the Bayes filter consists of the Data Update step, which incorporates the sensor observation at time $k+1$ via the sensor likelihood $f_k(\mathbf{z}_{k+1}|\mathbf{x})$, which is the likelihood that a target with state $\mathbf{x}$ will produce a sensor observation $\mathbf{z}_{k+1}$. The observation $\mathbf{z}_{k+1}$ is thus used to weight the prediction density produced by the Time Update step to obtain the final Bayesian posterior density for time step $k+1$:

$$f_{k+1|k+1}(\mathbf{x}|Z^{k+1}) = \frac{f_k(\mathbf{z}_{k+1}|\mathbf{x}) f_{k+1|k}(\mathbf{x}|Z^k)}{f_{k+1}(\mathbf{z}_{k+1}|Z^k)}, \tag{4}$$

where

$$f_{k+1}(\mathbf{z}_{k+1}|Z^k) = \int f_k(\mathbf{z}_{k+1}|\mathbf{x}) f_{k+1|k}(\mathbf{x}|Z^k) d\mathbf{x}$$

is a normalizing term. The target state at time $k+1$ can be extracted, as at time $k$, by applying (1) or (2) to the posterior density produced by (4).

## *2.2 Multitarget, Multisensor Tracking*

### 2.2.1 Random Sets and the Set Integral

In an attempt to solve the data fusion and multitarget tracking problems discussed in Section 1.1, Mahler proposes the use of "random finite sets," which map objects in a probability space to sets [43, 49, 57]. In this approach, all sensor observations are regarded as belonging to the random finite set $Z$, and the target states similarly belong to the random finite set $X$. Both $X$ and $Z$ contain a random number of randomly varying elements as shown in Table 1.

For example, in the case of no targets present, $X = \{\varnothing\}$; in the case of one target, $X = \{\mathbf{x}_1\}$; and in the case of two targets, $X = \{\mathbf{x}_1, \mathbf{x}_2\}$; and so forth, where $\mathbf{x}_i$ are the random target state vectors. The random set method thus reduces the multitarget, multisensor problem to a "single-target, single-sensor problem," where $X$ and $Z$ can be treated as the random "single" target and random "single" sensor observation, respectively.

**Table 1:** Random Finite Sets

- *Target state random finite set:*

  $X = \emptyset, \qquad X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}, \qquad n, \mathbf{x}_i \qquad r.v.s$

- *Sensor observation random finite set:*

  $Z = \emptyset, \qquad Z = \{\mathbf{z}_1, \ldots, \mathbf{z}_m\}, \qquad m, \mathbf{z}_i \qquad r.v.s$

Furthermore, the random set method provides a convenient way to incorporate false alarms, target births and deaths, and nonunity probabilities of detection. The observation random finite set, for example, can be treated as a union of random sets:

$$Z_k = O_k(X) \cup C_k, \tag{5}$$

where $O_k(X)$ consists of the random sensor observations modeled with some noise and probability of detection, and $C_k$ consists of the random false-alarm observations. Likewise, the random time-updated target state can be treated as the union of random sets:

$$X_{k+1|k} = D_k(X) \cup B_k, \tag{6}$$

where $D_k(X)$ consists of the random target states in $X$ predicted forward to time $k+1$, taking into account target spawning and disappearances, and $B_k$ consists of the random target states of new targets that spontaneously appear at time $k+1$.

Now that the concept of a random finite set has been defined, we can use it to extend the Bayes filter to the multitarget, multisensor case by using the multi-object equivalents of the sensor likelihood and target motion model likelihood, namely $f_k(Z|X)$ and $f_{k+1|k}(Y|X)$. To construct these likelihoods, the tools of finite-set statistics [43] are needed. Primarily of interest are the set integral and set derivative, which are given in Table 2. We focus now on the set integral and return to the set derivative to obtain the multiobject likelihoods in Section 2.2.2.

Given a function $f(X)$, its set integral with respect to the random set $X$ over some space $S$, where $S \supseteq X$, is denoted as

**Table 2:** FISST Set Integral and Set Derivative

- *Set Integral:*

  $\int f(X)\delta X = f(\emptyset) + \sum_{n=1}^{\infty} \frac{1}{n!} \int f(\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}) d\mathbf{x}_1 \cdots d\mathbf{x}_n$

- *Set Derivative:*

  $\frac{\delta F}{\delta \mathbf{x}}(S) = \lim_{\lambda(E_x)\to 0} \frac{F(S \cup E_x) - F(S)}{\lambda(E_x)}$

$$\begin{aligned}\int_S f(X)\delta X &= \sum_{n=0}^{\infty} \frac{1}{n!} \int_{S^n} f(\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}) d\mathbf{x}_1 \cdots d\mathbf{x}_n \\ &= f(\emptyset) + \int_S f(\{\mathbf{x}\}) d\mathbf{x} + \frac{1}{2} \int_{S \times S} f(\{\mathbf{x}_1, \mathbf{x}_2\}) d\mathbf{x}_1 d\mathbf{x}_2 + \ldots \end{aligned} \tag{7}$$

Note that $S$ is not a random finite set, but rather the space specifying the possible values that the elements of $X$ can take (i.e. $\mathbf{x}_i \in S$ and $X \subseteq S$). If the function $f(X)$ is a multiobject likelihood, then its set integral can be interpreted as being the sum of the individual probabilities of having $n$ objects in $S$, where $f(\{\mathbf{x}_1, \ldots, \mathbf{x}_n\})$ is the likelihood of the random set $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ occuring in $S$. In (7), the first term is the probability of having no objects in $S$, the second term is the probability of having a single object in $S$, the next term is the probability of having two objects in $S$, and so on. If the set integral is taken over the entire state space (and not just any subset $S$), then the multiobject likelihood should integrate to one, just like any probability density function. Thus, integrating the sensor and target motion likelihoods over their entire respective state spaces gives

$$\int f_k(Z|X)\delta Z = 1 \tag{8}$$

and

$$\int f_{k+1|k}(Y|X)\delta Y = 1. \tag{9}$$

To explain the $\frac{1}{n!}$ term in (7), it is worth taking a moment to point out the difference between set notation and vector notation [57]. In set notation, the order of the elements in the set does not matter. That is, $f(\{\mathbf{x}_1, \mathbf{x}_2\})$ specifies the likelihood that $\mathbf{x}_1$ and $\mathbf{x}_2$ exist. In

vector notation, $f(\mathbf{x}_1, \mathbf{x}_2)$ is the likelihood that the specific sequence $(\mathbf{x}_1, \mathbf{x}_2)$ exists, which is different from $f(\mathbf{x}_2, \mathbf{x}_1)$. Table 3 illustrates how the likelihood of a set is $n!$ times greater than the likelihood of a vector. This scale term is present in (7), since we are concerned only with the probability of the existence of the elements of $X$ in $S$, and not with the particular order of those elements.

**Table 3:** Set Notation and Vector Notation

$$f(\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}) = n!\, f(\mathbf{x}_1, \ldots, \mathbf{x}_n)$$

**Assumes all vector permutations are equally likely.*

*Example, three objects:*

$$\begin{aligned} f(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) &= 0.1 \qquad & f(\mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_1) &= 0.1 \\ f(\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_2) &= 0.1 \qquad & f(\mathbf{x}_3, \mathbf{x}_1, \mathbf{x}_2) &= 0.1 \\ f(\mathbf{x}_2, \mathbf{x}_1, \mathbf{x}_3) &= 0.1 \qquad & f(\mathbf{x}_3, \mathbf{x}_2, \mathbf{x}_1) &= 0.1 \end{aligned}$$

*and so,*

$$\begin{aligned} f(\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}) &= 3!\, f(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) \\ &= 6 \cdot (0.1) \end{aligned}$$

### 2.2.2 Belief-mass Functions and the Set Derivative

Although the PHD-based filter can be derived by directly using the probability generating functional (p.g.fl.) of point process theory[1] (see Section 2.2.3), it is worthwhile to review the concept of the belief-mass function promoted by Mahler, since it can be readily generalized to the p.g.fl. and provides a way to illustrate the concepts of the set integral and set derivative of FISST. As mentioned in Section 1.2, a detailed look at random set theory is provided in [21], and Mahler attempts to make the theory accessible by drawing parallels between the commonly known random variable concepts and those of random sets [43]. He

[1] Mori and Chong conclude [60] that the random set approach to multitarget tracking is equivalent to a point process approach with the exception that two targets cannot occupy the same state $x$. Hence, a random set is the same as a simple point process, which is a point process that has no duplicate elements. See [14] for more on point process theory.

introduces the concept of a belief-mass function, which behaves in an analogous fashion to the more commonly known probability mass function. That is, the belief-mass function of the random observation set is

$$\beta_k(S|X) = \Pr(Z_k \subseteq S) = \int_S f_k(Z|X)\delta Z, \tag{10}$$

and represents the probability that the observations at time $k$ are in the space $S$, given a target-state random set $X$. Likewise, the belief-mass function of the time-updated random target set is

$$\beta_{k+1|k}(S|X) = \Pr(X_{k+1|k} \subseteq S) = \int_S f_{k+1|k}(Y|X)\delta Y, \tag{11}$$

which is the probability that the time-updated target states in random set $X_{k+1|k}$ are contained in the space $S$.

As seen in (10) and (11), the multiobject likelihoods can be extracted by using the set derivative, which is the dual of the set integral, as demonstrated below [43, 47].

Consider $E_z$ to be a small, closed region containing $\mathbf{z}$. Then, using the Lebesgue measure $\lambda(E_\mathbf{z})$, the probability mass of the region $E_z$ is

$$F(E_\mathbf{z}|\mathbf{x}) = \int_{E_\mathbf{z}} f(\mathbf{y}|\mathbf{x})dy \;\cong\; f_k(\mathbf{z}|\mathbf{x})\lambda(E_\mathbf{z}). \tag{12}$$

Then,

$$f_k(\mathbf{z}|\mathbf{x}) \;\cong\; \frac{F(E_\mathbf{z}|\mathbf{x})}{\lambda(E_\mathbf{z})} \tag{13}$$

and

$$f_k(\mathbf{z}|\mathbf{x}) \;=\; \lim_{\lambda(E_z)\to 0} \frac{F(E_\mathbf{z}|\mathbf{x})}{\lambda(E_\mathbf{z})}, \tag{14}$$

which is the Radon-Nikodým derivative. Assuming $S \cap E_\mathbf{z} = \emptyset$ and the limit exists,

$$f_k(\mathbf{z}|\mathbf{x}) \;=\; \lim_{\lambda(E_z)\to 0} \frac{F(S \cup E_\mathbf{z}|\mathbf{x}) - F(S|\mathbf{x})}{\lambda(E_\mathbf{z})}, \tag{15}$$

which is the set derivative:

$$f_k(\mathbf{z}|\mathbf{x}) \;=\; \frac{\delta\beta}{\delta\mathbf{z}}(S|\{\mathbf{x}\}). \tag{16}$$

The multitarget sensor observation p.d.f. is then given by

$$f_k(Z|X) = \frac{\delta^m \beta_k}{\delta \mathbf{z}_1 \cdots \delta \mathbf{z}_m}(S|X)\Big|_{S=\emptyset}, \tag{17}$$

where $S$ is conveniently chosen to be $\emptyset$, so as to satisfy the assumption that $S \cap E_{\mathbf{z}} = \emptyset$.[2] Likewise, the multitarget motion model likelihood is given by

$$f_{k+1|k}(Y|X) = \frac{\delta^n \beta_{k+1|k}}{\delta \mathbf{y}_1 \cdots \delta \mathbf{y}_n}(S|X)\Big|_{S=\emptyset}. \tag{18}$$

To illustrate the concept, Table 4 refers to a simple two-target example given by Mahler [47].

[2]Mahler commonly writes his equations using a typical shorthand notation, in which (17) would be written as:

$$f_k(Z|X) = \frac{\delta^m \beta_k}{\delta \mathbf{z}_1 \cdots \delta \mathbf{z}_m}(\emptyset|X).$$

For clarity, we will employ the traditional calculus notation used in (17).

**Table 4:** Belief-mass Two Target Example (taken from a tutorial given by Mahler [47])

---

*Given two existing targets,*

$$X = \{\mathbf{x}_1, \mathbf{x}_2\},$$

*and assuming unity probability of detection and no false alarms, we obtain a set of two random-variable observations, the first due to target $x_1$, and the second to $x_2$:*

$$Z = \{\mathbf{z}_1\} \cup \{\mathbf{z}_2\}$$

*Then, the belief-mass function is*

$$\begin{aligned} \beta_Z(S|X) = Pr(Z \subseteq S) &= Pr(\{\mathbf{z}_1\} \cup \{\mathbf{z}_2\} \subseteq S) \\ &= Pr(\mathbf{z}_1 \in S) Pr(\mathbf{z}_2 \in S) \\ &= F(S|\mathbf{x}_1) F(S|\mathbf{x}_2), \end{aligned}$$

*which is the product of the probability mass functions, assuming conditional independence.*

*Taking the set derivative of the belief-mass function provides us with the multitarget likelihood:*

$$f_Z(\emptyset|X) = \frac{\delta \beta_Z}{\delta \emptyset}(S|X)\bigg|_{S=\emptyset} = \beta_Z(\emptyset|X) = F(\emptyset|\mathbf{x}_1)F(\emptyset|\mathbf{x}_2) = 0$$

$$f_Z(\{\mathbf{z}_1\}|X) = \frac{\delta \beta_Z}{\delta \mathbf{z}_1}(S|X)\bigg|_{S=\emptyset} = f(\mathbf{z}_1|\mathbf{x}_1)F(\emptyset|\mathbf{x}_2) + F(\emptyset|\mathbf{x}_1)f(\mathbf{z}_1|\mathbf{x}_2) = 0$$

$$f_Z(\{\mathbf{z}_2, \mathbf{z}_1\}|X) = \frac{\delta^2 \beta_Z}{\delta \mathbf{z}_2 \delta \mathbf{z}_1}(S|X)\bigg|_{S=\emptyset} = f(\mathbf{z}_1|\mathbf{x}_1)f(\mathbf{z}_2|\mathbf{x}_2) + f(\mathbf{z}_2|\mathbf{x}_1)f(\mathbf{z}_1|\mathbf{x}_2)$$

$$f_Z(\{\mathbf{z}_3, \mathbf{z}_2, \mathbf{z}_1\}|X) = \frac{\delta^3 \beta_Z}{\delta \mathbf{z}_3 \delta \mathbf{z}_2 \delta \mathbf{z}_1}(S|X)\bigg|_{S=\emptyset} = 0$$

*Note that the likelihood falls out of the derivative at the correct number of observations.*

---

Additional random set equivalents of single-target statistical tools, such as the chain rule and sum and product rules, are also defined in FISST. Thus, Mahler has been able to generate a multitarget, multisensor equivalent of single-target, single-sensor theory. Using FISST and the multitarget likelihoods, the multitarget-version of the Bayes filter can be constructed to perform multitarget, multisensor tracking, in which the full FISST posterior is propagated forward in time from one iteration to another, as shown in Table 5:

*Time Update step:*

$$f_{k+1|k}(X|Z^k) = \int f_{k+1|k}(X|W) f_{k|k}(W|Z^k) \delta W. \tag{19}$$

*Data Update step:*

$$f_{k+1|k+1}(X|Z^{k+1}) = \frac{f_k(Z^{k+1}|X) f_{k+1|k}(X|Z^k)}{f_{k+1}(Z^{k+1}|Z^k)}. \tag{20}$$

A multitarget, multisensor tracker using the full FISST posterior density has been implemented by Sidenbladh [69, 70].

**Table 5:** Multitarget Bayes Filter Diagram

| |
|---|
| $f_{k\|k}(X\|Z^k) \xrightarrow[Time\ Update]{} f_{k+1\|k}(X\|Z^k) \xrightarrow[Data\ Update]{} f_{k+1\|k+1}(X\|Z^{k+1})$ |

### 2.2.3 The Probability Hypothesis Density (PHD)

To be of use to target tracking engineers, the random set theory must be made as easy to use as the traditional Kalman style filters. Propagating the full FISST posterior density is cumbersome, as noted in [46, 69]. In the traditional Kalman filter, the filter recursion is performed on a manageably small set of sufficient statistics (namely, a mean and covariance) that characterize the prior and posterior target state distributions. Mahler attempts to derive an analogous set of sufficient statistics for the multitarget tracking filter. As an analogous statistic to the mean of the Kalman filter, Mahler [18, 46, 48, 53, 56] proposes

using the probability hypothesis density (PHD) as the first moment statistic propagated in an approximate multitarget, multisensor Bayes filter.[3]

To derive the PHD and the PHD-based Bayes filter, we now turn to the theory of point processes. Point process theory [14] is a framework in which to analyze the occurrences of events in time or of objects in a region. Point process theory contains the concept of the probability generating functional (p.g.fl.), which is analogous to the characteristic functions of single random variables. Given a state-space $S$ from which a random finite set $X$ can be drawn, and given a function $h(\cdot)$, such that $|h(\cdot)| \leq 1$, the p.g.fl. is given as

$$G[h] = \mathrm{E}\Big(\prod_i h(\mathbf{x}_i)\Big) = \int h^X \ f(X)\delta X, \tag{21}$$

which is the expectation of the function $h^X = \prod_i h(\mathbf{x}_i)$, where $\mathbf{x}_i$ are the elements of $X$, and where $h^X = 1$ if $X = \emptyset$. For instance, $h^{\{4,3,2\}}$ denotes $h(4) \cdot h(3) \cdot h(2)$. Using the p.g.fl. and its properties, one is able to construct the PHD.

The PHD, $\hat{D}_{k|k}(\mathbf{x}|Z^{(k)})$, is equivalent to the first-order factorial moment density in point process theory, and it is uniquely characterized by the property that when it is integrated over any region of state-space, it produces the expected[4] number of objects contained in that region:

$$N_{k|k}(S) = \int_S \hat{D}_{k|k}(\mathbf{x}|Z^{(k)})d\mathbf{x}. \tag{22}$$

Thus, in terms of multitarget tracking, the PHD provides a straightforward method of estimating the number of targets in a region under observation. For notational clarity, the $\hat{}$ symbol above a variable indicates that it is a PHD.

Mahler, in [48, 56], proves and summarizes some of the properties of the functional derivatives of the p.g.fl. He notes that the functional (or gradient) derivative is analogous

[3]Mahler proposed an extension of the Gaussian density to the multitarget case [55] that allows the computations in a FISST-based filter to be done in closed form. Although this was a first step toward thinking about better approaches, Mahler pointed out (p. 312 of [55]) that "only target disappearances (and not appearances) can be modeled... [and] a *single* covariance matrix $P_k$ is forced to describe the uncertainty in the state estimate of *every* target. Nevertheless, it is an initial foray that illustrates the value of a systematic mathematical foundation for multisensor-multitarget problems." This line of reasoning was later dropped in favor of the PHD.

[4]The term "expected" is used in the mathematical sense of the expected value of a random number, not in the sense of "supposed," "assumed," or "predicted." Recall that in the random set framework, the number of objects in the target-state set $X$ is a random variable.

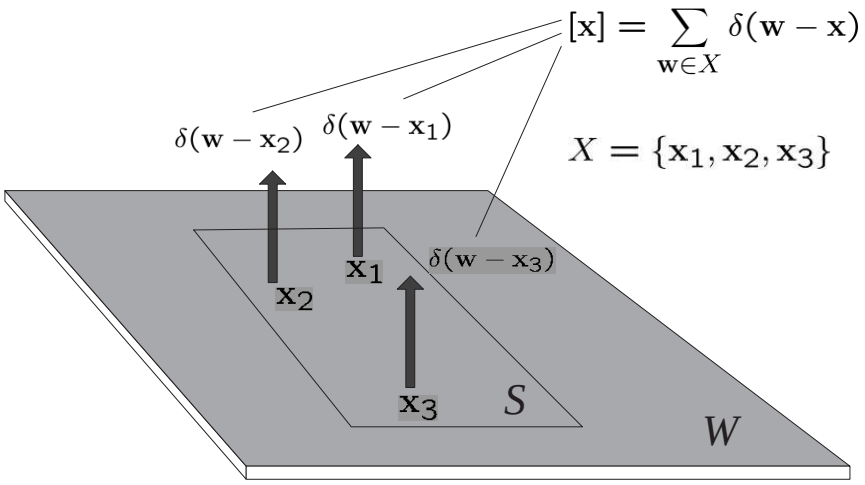



**Figure 1:** An example of a point process for the case in which three objects are present. The state space is indicated by $W$, which contains all possible target states $w$, where $w$ is a target state vector, e.g., $w = [x, y, \dot{x}, \dot{y}]^{\mathrm{T}}$ in which the target state consists of location and velocity information. The random set of target states is indicated by $X = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$, which contains the states of the targets present. The $\mathbf{x}_i \in W$ are the states of the three targets actually present. The point process $[\mathbf{x}] = \sum_{\mathbf{w}\in X} \delta(\mathbf{w}-\mathbf{x}) = \delta(\mathbf{w}-\mathbf{x}_1)+\delta(\mathbf{w}-\mathbf{x}_2)+\delta(\mathbf{w}-\mathbf{x}_3)$ is the sum of Dirac delta functions at the target states. $S$ is any subspace of $W$ (i.e., $S \subseteq W$) with which one is concerned. For example, $S$ may be the region of state space over which one integrates the PHD to find the number of targets present therein.

to the set derivative of FISST and proves that the PHD can be obtained from the p.g.fl. in the following manner:

$$\hat{D}(\{\mathbf{x}\}) \;=\; \left.\frac{\delta G[h]}{\delta \mathbf{x}}\right|_{h=1.} \tag{23}$$

To see why this is the case, consider the random point process $[\mathbf{x}] = \sum_{\mathbf{w}\in X} \delta(\mathbf{w}-\mathbf{x})$, where the objects in the random set $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ are represented as points (i.e., Dirac delta functions) in the space $W$.[5] Note that $n$ is random also. A realization of the point process is illustrated in Figure 1 for the case in which three targets are present. Taking the gradient derivative of the p.g.fl., we obtain:

$$\frac{\delta G}{\delta \mathbf{x}}[h] = \int \left(\frac{\delta}{\delta \mathbf{x}} h^X\right) f(X)\delta X. \tag{24}$$

Recalling that the PHD is the first moment of a point process, we would like to show that

$$\left.\frac{\delta}{\delta \mathbf{x}} h^X\right|_{h=\mathbf{1_S}} = \sum_{\mathbf{w}\in X} \delta(\mathbf{w}-\mathbf{x}), \tag{25}$$

[5]Mahler writes the Dirac delta function, $\delta(\mathbf{w}-\mathbf{x})$, as $\delta_{\mathbf{w}}(\mathbf{x})$. However, this notation is easily confused with a common set-membership notation, in which what we call $\mathbf{1_S}(\cdot)$ in (30) is often written as $\delta_{\mathbf{X}}(\cdot)$.

where $\mathbf{1_S}$ is the Dirac probability measure described below. Explicitly expanding the derivative term, and using the product rule for set derivatives [43], one obtains

$$\frac{\delta}{\delta \mathbf{x}} h^X = \frac{\delta}{\delta \mathbf{x}} \prod_i h(\mathbf{x})\Big|_{\mathbf{x}=\mathbf{x}_i} \tag{26}$$

$$= \frac{\delta}{\delta \mathbf{x}} \left[ h(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_1} \cdots h(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_n} \right] \tag{27}$$

$$= \left( \frac{\delta}{\delta \mathbf{x}} h(\mathbf{x})\Big|_{\mathbf{x}=\mathbf{x}_1} \right) h(\mathbf{x}_2) \cdots h(\mathbf{x}_n) + \cdots + h(\mathbf{x}_1) \cdots h(\mathbf{x}_{n-1}) \left( \frac{\delta}{\delta \mathbf{x}} h(\mathbf{x})\Big|_{\mathbf{x}=\mathbf{x}_n} \right) \tag{28}$$

$$= \sum_{i=1}^{n} h(\mathbf{x}_1) \cdots \left( \frac{\delta}{\delta \mathbf{x}} h(\mathbf{x})\Big|_{\mathbf{x}=\mathbf{x}_i} \right) \cdots h(\mathbf{x}_n). \tag{29}$$

If we let $h(\cdot)$ be the Dirac probability measure, $\mathbf{1_S}(\cdot)$, of the elements of $X$ existing in some region $S$ of the state space $W$, i.e., $S \subseteq W$ and

$$h(\mathbf{x}_i) = \mathbf{1_S}(\mathbf{x}_i) = \begin{cases} 1, \ \mathbf{x}_i \in S \\ 0, \ \mathbf{x}_i \notin S, \end{cases}$$

then

$$\frac{\delta}{\delta \mathbf{x}} h(\mathbf{x})\Big|_{\mathbf{x}=\mathbf{x}_i} = \delta(\mathbf{x} - \mathbf{x}_i) \tag{30}$$

is the Dirac delta function [48]. Recall that the set derivative of a probability measure is its density function.

Continuing with (29),

$$\frac{\delta}{\delta \mathbf{x}} h^X = \sum_{i=1}^{n} h(\mathbf{x}_1) \cdots \left( \frac{\delta}{\delta \mathbf{x}} h(\mathbf{x})\Big|_{\mathbf{x}=\mathbf{x}_i} \right) \cdots h(\mathbf{x}_n) \tag{31}$$

$$= \sum_{i=1}^{n} h(\mathbf{x}_1) \cdots \delta(\mathbf{x} - \mathbf{x}_i) \cdots h(\mathbf{x}_n) \tag{32}$$

$$= \sum_{i=1}^{n} (1) \cdots \delta(\mathbf{x} - \mathbf{x}_i) \cdots (1) \tag{33}$$

$$= \sum_{\mathbf{w} \in X} \delta(\mathbf{x} - \mathbf{w}) \tag{34}$$

Thus, we see that

$$\frac{\delta G}{\delta \mathbf{x}}[h] = \int \left( \sum_{\mathbf{w} \in X} \delta(\mathbf{w} - \mathbf{x}) \right) \ f(X) \delta X \tag{35}$$

is the first moment of a point process when the function $h$ is the function $\mathbf{1_S}$, and the PHD is

$$\hat{D}(\{\mathbf{x}\}) \ = \ \frac{\delta G[h]}{\delta \mathbf{x}}\Big|_{h=\mathbf{1_S}}. \tag{36}$$

In the FISST framework, Mahler introduces a p.g.fl. with $h = \mathbf{1_S}$ and calls this the belief-mass function with which to tackle the target tracking problem. He then generalizes this constraint and shows (p. 1170, [56]), in a similar fashion to the steps above, that the multitarget moment densities of a random set can be found by setting $h = 1$. That is, the $n$-th multitarget moment density is given by:

$$D(\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}) \;=\; \left.\frac{\delta^n G[h]}{\delta \mathbf{x}_1 \cdots \delta \mathbf{x}_n}\right|_{h=1.} \tag{37}$$

So, the first-order moment density (i.e., the PHD) is given by:

$$\hat{D}(\{\mathbf{x}\}) \;=\; \left.\frac{\delta G[h]}{\delta \mathbf{x}}\right|_{h=1.} \tag{38}$$

For ease of notation, we will use $\hat{D}(\mathbf{x})$ to denote the PHD.

In summary, the PHD is defined as being any density that, when integrated over a region of state space, provides the expected number of objects in that region; see (22). More specifically, the PHD is the first-order factorial moment of a point process:

$$\hat{D}(\mathbf{x}) = \hat{D}(\{\mathbf{x}\}) = \int f(\{\mathbf{x}\} \cup W)\delta W \tag{39}$$

$$= \int [\mathbf{x}] \cdot f(X)\delta X, \;\; \left(\text{where } \; [\mathbf{x}] = \sum_{\mathbf{w} \in X} \delta(\mathbf{w} - \mathbf{x})\right) \tag{40}$$

$$= \int \left(\sum_{\mathbf{w} \in X} \delta(\mathbf{w} - \mathbf{x})\right) \; f(X)\delta X. \tag{41}$$

Expanding the set integral of (40), and noting that the $[\mathbf{x}]$ term requires that $\mathbf{x}$ be one of the objects in $X$, gives

$$\hat{D}(\mathbf{x}) = \sum_{n=0}^{\infty} \frac{1}{n!} \int [\mathbf{x}]\, f(\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}) d\mathbf{x}_1 \cdots d\mathbf{x}_n \tag{42}$$

$$= \sum_{n=0}^{\infty} \frac{1}{n!} \int f(\{\mathbf{x}, \mathbf{x}_1, \ldots, \mathbf{x}_n\}) d\mathbf{x}_1 \cdots d\mathbf{x}_n \tag{43}$$

$$= f(\{\mathbf{x}\}) + \int f(\{\mathbf{x}, \mathbf{x}_1\}) d\mathbf{x}_1 \; + \frac{1}{2} \int f(\{\mathbf{x}, \mathbf{x}_1, \mathbf{x}_2\}) d\mathbf{x}_1 d\mathbf{x}_2 \; + \; \ldots \tag{44}$$

When integrated over a region $S$, the first term in (44) is the probability that there is only one target in the region and that the target has state $\mathbf{x}$ in $S$. The second term, when integrated over $S$ with respect to $\mathbf{x}$, is the probability that there are two targets in $S$ and

that one of them has state $\mathbf{x}$. Likewise, the $n$-th term in the summation, when integrated over $S$ with respect to $\mathbf{x}$, is the probability that there are $n$ targets in $S$ and that one of the targets has state $\mathbf{x}$.

As noted by Erdinc [19], the $n$-th term in the summation of (44) is a sum of the marginal densities of there being a target with state $\mathbf{x}$, assuming that there are $n$ targets present, weighted by the probability of there being $n$ targets. Thus, for the $(n+1)$-th term of (44),

$$\frac{1}{n!}\int f(\{\mathbf{x},\mathbf{x}_1,\ldots,\mathbf{x}_n\})d\mathbf{x}_1\cdots d\mathbf{x}_n = \frac{1}{n!}\int (n+1)!f(\mathbf{x},\mathbf{x}_1,\ldots,\mathbf{x}_n)d\mathbf{x}_1\cdots d\mathbf{x}_n \quad (45)$$

$$= (n+1)\int f(\mathbf{x},\mathbf{x}_1,\ldots,\mathbf{x}_n)d\mathbf{x}_1\cdots d\mathbf{x}_n \quad (46)$$

$$= (n+1)\cdot p_{n+1}\int \Pi(\mathbf{x},\mathbf{x}_1,\ldots,\mathbf{x}_n)d\mathbf{x}_1\cdots d\mathbf{x}_n, \quad (47)$$

where in (45), we naturally assume all permutations of the vector $(\mathbf{x},\mathbf{x}_1,\ldots,\mathbf{x}_n)$ to be equally likely (as described in Section 2.2.1), and where in (47), $p_{n+1}$ is the probability of there being $n+1$ targets, and $\Pi(\mathbf{x},\mathbf{x}_1,\ldots,\mathbf{x}_n)$ is the joint probability density for the vector of target states $(\mathbf{x},\mathbf{x}_1,\ldots,\mathbf{x}_n)$. Note that the joint probability density has the same value for all vector permutations of any particular $n+1$ target states.

Thus, the PHD at a point $\mathbf{x}$ in state space is the sum of the infinitesimal probabilities[6] of all target number scenarios for which $\mathbf{x}$ is the state of a target. Note, however, that the PHD is not a multitarget p.d.f. and does not integrate to one, but instead integrates to the expected number of targets present in the region $S$ under consideration:

$$N_{k|k}(S) = \int_S \hat{D}(\mathbf{x})d\mathbf{x}. \quad (48)$$

### 2.2.4 Multiobject Time-Update and Data-Update Equations

Now that the concept of the probability hypothesis density has been introduced, we need to use the multitarget, multisensor Bayes filter equations ((19) and (20)) to propagate the PHD forward in time, i.e. from $\hat{D}_{k|k}(\mathbf{x}|Z^{(k)})$ to $\hat{D}_{k+1|k+1}(\mathbf{x}|Z^{(k+1)})$.

[6] By "infinitesimal," we mean that one must integrate the PHD over a small region in the state-space about the point $\mathbf{x}$ to obtain the probability at $\mathbf{x}$. We avoid using the term "density" in place of "infinitesimal probability," since the usual definition of a density implies that it integrates to one when integrated over the entire state space. However, the terms of the PHD equation (in (44)) do not necessarily integrate to one, either as a whole or individually, since each term is implicitly weighted by the probability of there being the number of targets assumed present by each term. Recall that the $n$-th term of (44) assumes that there are $n$ targets present.

#### 2.2.4.1 Time Update

Let us revisit the multitarget Time Update equation:

$$f_{k+1|k}(X|Z^k) = \int f_{k+1|k}(X|W) f_{k|k}(W|Z^k) \delta W. \tag{49}$$

According to (23), the corresponding time-updated PHD, $\hat{D}_{k+1|k}(\mathbf{x})$, can be found via

$$\hat{D}_{k+1|k}(\mathbf{x}) \;=\; \left. \frac{\delta G_{k+1|k}[h]}{\delta \mathbf{x}} \right|_{h=1,} \tag{50}$$

where $G_{k+1|k}[h]$ is the probability generating functional (p.g.fl.) of $f_{k+1|k}(X|Z^k)$.

Mahler [48] constructs the p.g.fl. in the following manner:

$$G_{k+1|k}[h] = \int h^X \; f_{k+1|k}(X|Z^k) \delta X \tag{51}$$

$$= \int \left\{ \int h^X \; f_{k+1|k}(X|W) \delta X \right\} \; f_{k|k}(W|Z^k) \delta W \tag{52}$$

$$= \int G_{k+1|k}[h|X] \; f_{k|k}(X|Z^k) \delta X, \tag{53}$$

where one now needs to find $G_{k+1|k}[h|X]$, which is the p.g.fl. of the multitarget motion model likelihood $f_{k+1|k}(X|W)$. Mahler achieves this by constructing the random set $X_{k+1|k}$, which is the result of applying the target motion model and which accounts for the predicted target states after the Bayes filter prediction step. The property that the p.g.fl. of a union of independent random sets is the product of the p.g.fls. of the random sets [14, 48] is also exploited.

In the random set framework (see (6)) proposed by Mahler, the time-updated target-state random set $X_{k+1|k}$ is represented as the union of the following independent sets:

$$X_{k+1|k} \;=\; T(X) \;\cup\; \Psi(X) \;\cup\; \Psi_0, \tag{54}$$

where

$$T(X) = T(\mathbf{x}_1) \cup \ldots \cup T(\mathbf{x}_n) \tag{55}$$

is the set of targets that survive from time $k$ to time $k+1$;

$$\Psi(X) = \Psi(\mathbf{x}_1) \cup \ldots \cup \Psi(\mathbf{x}_n) \tag{56}$$

is the set of targets that exist at time $k+1$, having been spawned from targets that exist at time $k$; and $\Psi_0$ is the set of birth targets, which are targets that spontaneously appear at time $k+1$. Thus, $G_{k+1|k}[h|X]$ is the product of the p.g.fls. of the independent sets:

$$G_{k+1|k}[h|X] \;=\; G_T[h|\mathbf{x}_1]\cdots G_T[h|\mathbf{x}_n]\cdot G_\Psi[h|\mathbf{x}_1]\cdots G_\Psi[h|\mathbf{x}_n]\cdot G_{\Psi_0}[h]. \tag{57}$$

The p.g.fl. of a single surviving target, $G_T[h|\mathbf{x}_i]$, taking into account its probability of survival, $p_{S,i}$, and its Markov model target motion probability distribution, $f_{k+1|k,i}(\mathbf{y}|\mathbf{x}_i)$, is given as (p. 1164, [56]):

$$G_T[h|\mathbf{x}_i] = \int h^Y f_{k+1|k,i}(Y|\mathbf{x}_i)\delta Y \;=\; f(\emptyset) + \int h(\mathbf{y}) f_{k+1|k,i}(\{\mathbf{y}\}|\mathbf{x}_i)d\mathbf{y} \tag{58}$$

$$= 1 - p_{S,i} + \int h(\mathbf{y}) p_{S,i} f_{k+1|k,i}(\mathbf{y}|\mathbf{x}_i)d\mathbf{y} \tag{59}$$

$$= 1 - p_{S,i} + p_{S,i}p[h], \tag{60}$$

where

$$p[h] = \int h(\mathbf{y}) f_{k+1|k,i}(\mathbf{y}|\mathbf{x}_i)d\mathbf{y}. \tag{61}$$

Note that $1 - p_{S,i}$ is the probability that the target does not survive, and that $p_{S,i}p[h]$ is a weighted probability that the target does survive, weighted pointwise by the function $h$. Continuing with this example to illustrate the concept of the time-updated PHD, we use (23) and the property (p. 1162, [56]) that

$$\frac{\delta G[h]}{\delta \mathbf{x}} = f(\mathbf{x}), \text{ if } G[h] = \int h(\mathbf{x}) f(\mathbf{x}) d\mathbf{x}, \tag{62}$$

to obtain the PHD for this single target case:

$$\left.\frac{\delta G_T[h|\mathbf{x}_i]}{\delta \mathbf{y}}\right|_{h=1} = p_{S,i} f_{k+1|k,i}(\mathbf{y}|\mathbf{x}_i). \tag{63}$$

In this single target case, which takes into account only the probability of survival and target motion likelihood, the PHD is equivalent to the motion likelihood weighted by the probability of survival. Note that the PHD is not a multitarget likelihood, i.e., it does not integrate to one, but rather to the expected number of targets present. Expanding the example to a simple multitarget case makes this a little more evident. The p.g.fl. of $T(X)$,

the union of the sets of independently moving targets, each with probability of survival $p_{S,i}$ and target motion likelihood $f_{k+1|k,i}(\mathbf{y}|\mathbf{x})$, is

$$G_T[h|X] = G_T[h|\mathbf{x}_1] \cdots G_T[h|\mathbf{x}_n]. \tag{64}$$

Using the product rule [43, 56] of the set derivative,

$$\frac{\delta G_T[h|X]}{\delta \mathbf{y}} = \sum_{1 \leq i \leq n} G_T[h|\mathbf{x}_1] \cdots p_{S,i} f_{k+1|k,i}(\mathbf{y}|\mathbf{x}_i) \cdots G_T[h|\mathbf{x}_n]. \tag{65}$$

To obtain the PHD, we set $h = 1$ and note that the single target p.g.fl., as given in (60), then becomes

$$\begin{aligned} G_T[1|\mathbf{x}_i] &= 1 - p_{S,i} + p_{S,i} p[1] && (66)\\ &= 1 - p_{S,i} + p_{S,i} \int 1 \cdot f_{k+1|k,i}(\mathbf{y}|\mathbf{x}_i) d\mathbf{y} && (67)\\ &= 1 - p_{S,i} + p_{S,i} \cdot 1 && (68)\\ &= 1. && (69) \end{aligned}$$

So, the PHD of the multitarget set is

$$\begin{aligned} \hat{D}_T(\mathbf{y}) = \frac{\delta G_T[h|X]}{\delta \mathbf{y}}\bigg|_{h=1} &= \sum_{1 \leq i \leq n} G_T[1|\mathbf{x}_1] \cdots p_{S,i} f_{k+1|k,i}(\mathbf{y}|\mathbf{x}_i) \cdots G_T[1|\mathbf{x}_n] && (70)\\ &= \sum_{1 \leq i \leq n} 1 \cdots p_{S,i} f_{k+1|k,i}(\mathbf{y}|\mathbf{x}_i) \cdots 1 && (71)\\ &= \sum_{1 \leq i \leq n} p_{S,i} f_{k+1|k,i}(\mathbf{y}|\mathbf{x}_i) && (72)\\ &= p_{S,1} f_{k+1|k,1}(\mathbf{y}|\mathbf{x}_1) + \cdots + p_{S,n} f_{k+1|k,n}(\mathbf{y}|\mathbf{x}_n). && (73) \end{aligned}$$

Taking the integral of the PHD provides the expected number of targets present:

$$\begin{aligned} N_T &= \int \hat{D}_T(\mathbf{y}) d\mathbf{y} && (74)\\ &= \int \Big( p_{S,1} f_{k+1|k,1}(\mathbf{y}|\mathbf{x}_1) + \cdots + p_{S,n} f_{k+1|k,n}(\mathbf{y}|\mathbf{x}_n) \Big) d\mathbf{y} && (75)\\ &= p_{S,1} \int f_{k+1|k,1}(\mathbf{y}|\mathbf{x}_1) d\mathbf{y} + \cdots + p_{S,n} \int f_{k+1|k,n}(\mathbf{y}|\mathbf{x}_n) d\mathbf{y} && (76)\\ &= p_{S,1} + \cdots + p_{S,n}. && (77) \end{aligned}$$

In the case of this simple target motion model $T(X)$, the expected number of targets is just the sum of each target's probability of survival. So, for example, in the case where there

are ten targets, and each target has a probability of survival of 0.5, then five targets are expected to exist at the next time step, given that we use the random set $T(X)$ to model the target motion.

Returning to the complete target prediction model of (57), the p.g.fl. can be constructed as:

$$G_{k+1|k}[h|X] = G_T[h|\mathbf{x}_1] \cdots G_T[h|\mathbf{x}_n] \cdot G_\Psi[h|\mathbf{x}_1] \cdots G_\Psi[h|\mathbf{x}_n] \cdot G_{\Psi_0}[h] \tag{78}$$

$$= (1 - p_{S,1} + p_{S,1}p_1[h]) \cdots (1 - p_{S,n} + p_{S,n}p_n[h]) \cdot b_1[h] \cdots b_n[h] \cdot e[h], \tag{79}$$

where

$$1 - p_{S,i} + p_{S,i}p_i[h] \tag{80}$$

is the p.g.fl., as given in (60), for an individual target with probability of survival $p_{S,i}$ and Markov motion model likelihood density $f_{k+1|k,i}(\mathbf{y}|\mathbf{x})$, and

$$b_i[h] = \int h^Y \; b_{k+1|k}(Y|\mathbf{x}_i)\delta Y \tag{81}$$

is the p.g.fl. of $b_{k+1|k}(Y|\mathbf{x}_i)$, which is the likelihood that a target with state $\mathbf{x}_i$ at time $k$ will spawn the set of targets $Y$ at time $k+1$, and

$$e[h] = \int h^Y \; b_{k+1|k}(Y)\delta Y \tag{82}$$

is the p.g.fl. of $b_{k+1|k}(Y)$, which is the likelihood of spontaneous target births at time $k+1$.

Using $G_{k+1|k}[h|X]$, Mahler derives the Time Update equation for propagating the PHD from the posterior at time $k$, $\hat{D}_{k|k}(\mathbf{x}|Z^{(k)})$, to the predicted PHD at time $k$+1, $\hat{D}_{k+1|k}(\mathbf{y}|Z^{(k)})$, in the multitarget Bayes filter:

$$\hat{D}_{k+1|k}(\mathbf{y}|Z^{(k)}) = \; \hat{b}_{k+1|k}(\mathbf{y}) + \int \left( s_{k+1|k}(\mathbf{x}) f_{k+1|k}(\mathbf{y}|\mathbf{x}) + \hat{b}_{k+1|k}(\mathbf{y}|\mathbf{x}) \right) \hat{D}_{k|k}(\mathbf{x}|Z^{(k)}) d\mathbf{x}, \tag{83}$$

where $f_{k+1|k}(\mathbf{y}|\mathbf{x})$ is the single-target motion model's Markov transition density; $s_{k+1|k}(\mathbf{x})$ is the probability that the target at state $\mathbf{x}$ at time $k$ will survive to time $k+1$; $\hat{b}_{k+1|k}(\mathbf{y}|\mathbf{x})$ is the PHD of $b_{k+1|k}(Y_S|\mathbf{x})$, which is the multitarget likelihood density function of a target with state $\mathbf{x}$ at time $k$ spawning a set of new targets $Y_S$ at time $k+1$; and $\hat{b}_{k+1|k}(\mathbf{y})$ is the PHD of $b_{k+1|k}(Y_B)$, which is the likelihood function that a set of new targets $Y_B$ will

be born spontaneously at time $k+1$. The full derivation of the PHD-based Bayes filter time-update step is found in [48, 56].

#### *2.2.4.2 Data Update*

In a similar fashion, Mahler derives the Data Update equation for the PHD-based multitarget, multisensor Bayes filter. This time, a random set is used to model the sensor observations obtained at time $k+1$, and, like the time-updated target-state random set in (54) used to derive the Time Update equation, it is constructed to be the union of independent sets (see (5)):

$$Z_{k+1} = O_{k+1}(\mathbf{x}_1) \cup \cdots \cup O_{k+1}(\mathbf{x}_n) \cup C_{k+1}, \tag{84}$$

where $O_{k+1}(\mathbf{x}_i)$ is the observation caused by the target at state $\mathbf{x}_i$, and $C_{k+1}$ is the set containing the Poisson-distributed false alarms. It is assumed that the observations and the clutter are all statistically independent for a given multitarget state $X$, and so the following p.g.fl. of the observation model is constructed:

$$G_{k+1}[g|X] \;=\; G_O[g|\mathbf{x}_1] \cdots G_O[g|\mathbf{x}_n] \cdot G_C[g], \tag{85}$$

where

$$G_O[g|\mathbf{x}] = 1 - p_D(\mathbf{x}) + p_D(\mathbf{x}) p_g(\mathbf{x}) \tag{86}$$

is the p.g.fl. of a single observation, $p_D(\mathbf{x})$ is the probability of detection of a target with state $\mathbf{x}$, and

$$p_g(\mathbf{x}) = \int g(\mathbf{z}) f(\mathbf{z}|\mathbf{x}) d\mathbf{z}, \tag{87}$$

where $f(\mathbf{z}|\mathbf{x})$ is the likelihood of receiving an observation $\mathbf{z}$ given a target at state $\mathbf{x}$. Note the similarity to (58) - (61). Since the clutter model assumes that the false alarms are independent and Poisson-distributed, the p.g.fl. is [56]

$$G_C[g] = e^{\lambda \kappa[g] - \lambda}, \tag{88}$$

and

$$\kappa[g] = \int g(\mathbf{z}) c(\mathbf{z}) d\mathbf{z}, \tag{89}$$

where $\lambda$ is the Poisson false-alarm parameter, and $c(\mathbf{z})$ is the false-alarm density in the observation space. Details on the definition of these parameters and their selection are given in Section 4.4.4.

Using $G_{k+1}[g|X]$ and simplifying the derivation to make it practical by assuming that the predicted p.g.fl. from the Time Update step is approximately Poisson, i.e.,

$$G_{k+1|k}[h] = e^{\mu\sigma[h]-\mu}, \tag{90}$$

where

$$\sigma[h] = \int h(\mathbf{x})s(\mathbf{x})d\mathbf{x}, \tag{91}$$

for some probability density $s(\mathbf{x})$ where $\hat{D}_{k+1|k}(\mathbf{x}|Z^{(k)}) = \mu s(\mathbf{x})$, Mahler derives an approximate Data Update equation [48, 56] that incorporates the sensor observations into the time-update predicted PHD, $\hat{D}_{k+1|k}(\mathbf{x}|Z^{(k)})$, to obtain the posterior PHD at time $k+1$:

$$\hat{D}_{k+1|k+1}(\mathbf{x}|Z^{(k+1)}) \;\cong\; \left(\alpha_0 \;+ \sum_{\mathbf{z}_i \in Z_{k+1}} \alpha_i\right) \hat{D}_{k+1|k}(\mathbf{x}|Z^{(k)}), \tag{92}$$

where

$$\alpha_0 = 1 - p_D(\mathbf{x}) \tag{93}$$

and

$$\alpha_i = \frac{p_D(\mathbf{x})f(\mathbf{z}_i|\mathbf{x})}{\lambda_{k+1}c_{k+1}(\mathbf{z}_i) \;+ \int\; p_D(\mathbf{x})f(\mathbf{z}_i|\mathbf{x})\hat{D}_{k+1|k}(\mathbf{x}|Z^{(k)})d\mathbf{x}}\,. \tag{94}$$

$Z_{k+1} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_m\}$ is the set of observations collected by the sensor at time $k+1$, $f(\mathbf{z}|\mathbf{x})$ is the single-target likelihood function of the sensor reporting the observation, and the probability of detection is given by $p_D(\mathbf{x})$. The false-alarm rate is assumed to be Poisson distributed with parameter $\lambda_{k+1}$ and density $c_{k+1}(\mathbf{z})$. The full derivation of the posterior PHD for the Bayes filter data-update step is found in [48, 56].

Note that in this derivation of the posterior PHD, a single sensor has been assumed, since the probabilities of detection, sensor likelihood function, and clutter model do not identify which sensor is being used, as would be necessary in the case of multiple sensors. Mahler addresses this issue and shows how a rigorous multisensor derivation is impractical (p. 1169, [56]). This causes some undesired behavior in the PHD filter; we discuss this issue further, along with the nature of the approximation in (92), in Section 4.3.4.

#### *2.2.4.3 Estimating Target Number*

The expected number of targets at time $k+1$, which forms our estimate of the number of targets, is found by integrating the data-updated PHD and rounding to the nearest integer. That is,

$$N_{k+1|k+1} = \int \hat{D}_{k+1|k+1}(\mathbf{x}|Z^{(k+1)})dx \tag{95}$$

and

$$\tilde{N} = E[no.\ of\ targets] = [N_{k+1|k+1}]_{nearest\ integer.} \tag{96}$$

As a multitarget analog to the MAP estimator in (1), the location of the targets is found by extracting the location of the $\tilde{N}$ highest peaks of the data-updated PHD. More discussion on peak extraction is provided in Sections 4.3.5 and 5.2.4.

### 2.2.5 Conclusions

The Bayesian PHD filter implicitly performs two difficult tasks. The first task is that of estimating the number of targets present. This saves the tracking engineer from having to estimate the number of targets explicitly at each time step, which is a necessary step usually incorporated by ad-hoc means in other multitarget tracking algorithms. The second task is that of incorporating the sensor measurements without performing any explicit report-to-track association as mentioned in Section 1.1. These two features of the PHD filter make it an appealing tool to perform multitarget tracking in the context of passive radar, as described in Section 4.3.1, or when using multiple types of sensors and needing to fuse different kinds of observation data.

A summary of the PHD-based Bayes filter equations are provided in Table 6.

## *2.3 Other Applications of FISST* [7]

Although this chapter focused entirely on using FISST to track multiple point targets, Mahler's theory has found application in other areas. To help make this thesis serve as a centralized resource for the further study of FISST, we mention a few of them before moving on to the next chapter.

[7] This section was derived from unpublished, undated notes by Aaron Lanterman.

### 2.3.1 Higher-Level Sensor Fusion and Automatic Target Recognition

A wide variety of sensor fusion paradigms (such as fuzzy logic and Dempster-Shafer reasoning for imprecise evidence [40], as employed for theater-level fusion in [22] and SAR image analysis [29]) can be placed in the FISST framework. The notions are widely applicable; for instance, Mahler has sketched out an approach to tracking and classifying *groups* of targets for "situation assessment" or "force aggregation" [54].[8] Mahler has also extended [44] the "covariance intersection" concept of Uhlmann and Julier and the distributed tracking approach of Chong, Mori, and Chang [11]. The detection and acquisition of dim targets is discussed in [51]. "Soft" knowledge not readily represented in terms of probabilities can be incorporated using "conditional and relational event algebras," as discussed in Section III of [21]. A "fuzzy MAP estimator," derived from first principles from random set theory, has been applied to automatic target recognition with SAR imagery [24].

### 2.3.2 Sensor Resource Management

In [41], Mahler uses FISST to transform multitarget, multisensor management problems into conventional nonlinear optimal control problems and discusses the possibility of distributed sensors. Although the particular application involved is to determine optimal dwell allocations and waveform selection for radar systems, the underlying philosophy could be applied to any sort of sensor suite. In another application, Mahler and Prasanth [52] consider managing swarms of unmanned arial vehicles (UAVs), where the UAVs share information and also coordinate thier efforts via ideas from FISST.

### 2.3.3 Performance Bounds for Multitarget Tracking

Upon close examination, the concept of a "multitarget Cramér-Rao (CR) bound" is a bit nebulous. Strictly speaking, a CR bound is a matrix bound on covariances of random vectors. Hence, the set-theoretic CR bound in Section 5.3.2 of [21] is defined for vector-valued functions of the set parameters, and not on the set parameters themselves. For a known number of targets, it is not too hard to craft a reasonable performance metric.

[8] Also sometimes called "Level 2 fusion" in the JDL (Joint Directors of Laboratories) nomenclature.

But when the number of targets is unknown, the situation becomes trickier. For instance, how much weight should be put on missed targets or extraneous estimated targets versus errors in position estimation? Are a few false alarms worse than a systematic position error on a particular target? To begin formulating answers to these questions, Hoffman and Mahler [25], have proposed several metrics on target sets, such as Hausdorff distances and multi-object Wasserstein distances. With precise metrics defined, we have a yardstick for measuring the performance of existing algorithms. Hausdorff metrics, Hellinger measures, and multitarget Kullback-Leibler distances [42, 58] are used as performance measures for target tracking systems in [81].

**Table 6:** PHD-Based Bayesian Filter Equations: Summary

$$\hat{D}_{k|k}(\mathbf{x}|Z^{(k)}) \xrightarrow[Time\ Update]{} \hat{D}_{k+1|k}(\mathbf{x}|Z^{(k)}) \xrightarrow[Data\ Update]{approx.} \hat{D}_{k+1|k+1}(\mathbf{x}|Z^{(k+1)})$$

*Time Update:*

$$\hat{D}_{k+1|k}(\mathbf{x}|Z^{(k)}) = \hat{b}_{k+1|k}(\mathbf{x}) + \int \left(s_{k+1|k}(\mathbf{w}) f_{k+1|k}(\mathbf{x}|\mathbf{w}) + \hat{b}_{k+1|k}(\mathbf{x}|\mathbf{w})\right) \hat{D}_{k|k}(\mathbf{w}|Z^{(k)}) d\mathbf{w}$$

$f_{k+1|k}(\mathbf{x}|\mathbf{w})$ : single target motion likelihood
$s_{k+1|k}(\mathbf{w})$ : probability of survival
$\hat{b}_{k+1|k}(\mathbf{x}|\mathbf{w})$ : PHD of multitarget spawn likelihood
$\hat{b}_{k+1|k}(\mathbf{x})$ : PHD of multitarget birth likelihood

*Data Update:*

$$\hat{D}_{k+1|k+1}(\mathbf{x}|Z^{(k+1)}) \cong \left(\alpha_0 + \sum_{\mathbf{z}_i \in Z_{k+1}} \alpha_i\right) \hat{D}_{k+1|k}(\mathbf{x}|Z^{(k)}),$$

$$\alpha_0 = 1 - p_D(\mathbf{x}),$$

$$\alpha_i = \frac{p_D(\mathbf{x}) f(\mathbf{z}_i|\mathbf{x})}{\lambda_{k+1} c_{k+1}(\mathbf{z}_i) + \int p_D(\mathbf{x}) f(\mathbf{z}_i|\mathbf{x}) \hat{D}_{k+1|k}(\mathbf{x}|Z^{(k)}) d\mathbf{x}}$$

$f(\mathbf{z}_i|\mathbf{x})$ : single observation likelihood
$p_D(\mathbf{x})$ : probability of detection
$\lambda_{k+1}$ : Poisson-distributed false-alarm parameter
$c_{k+1}(\mathbf{z}_i)$ : false-alarm density

# CHAPTER III

# PARTICLE FILTERS

## *3.1 Introduction*

To tackle a variety of multitarget, multisensor tracking scenarios, particle filter implementations of the Bayesian PHD filter have been used [12, 13, 68, 72–74, 76, 77, 82, 83]. Particle filtering has also been used to implement a FISST filter that propagates the multitarget densities, not just the PHD approximation [69, 70]. An attractive feature of particle filters is that they are able to model nonlinear and non-Gaussian characteristics. They are also more tractable and flexible than grid-based methods, especially when the number of states in the system is infinite or when the state vectors are of high dimensionality. This makes them suitable for use when tracking with the PHD, since the target motion and observation models involved may be nonlinear and non-Gaussian. It has also been established that, as the number of particles is increased, the particle filter representation of the PHD converges to the true PHD for every stage of the PHD filter algorithm [32, 77].

In the case in which the motion and observation models are linear and Gaussian and target births are Gaussian, then a closed-form solution of the PHD filter as a mixture of Gaussians has been derived by Vo *et al.* in [75]. However, they note that the Gaussian mixture must be constantly pruned so as to make it computationally tractable. They also suggest methods to extend it to slightly nonlinear cases by using the approximation approaches from the extended Kalman filter and unscented Kalman filter. However, for generality and ease of use, we use a particle filter implementation of the Bayesian PHD filter equations in our simulations. To our knowledge, the first implementations of the PHD represented the posterior PHD using a truncation expansion (in some basis, such as a Fourier or wavelet basis) of the log-PHD [18]. This series-expansion approach was rapidly abandoned in favor of particle filters.

For the reader interested in the details of particle filters, there are many good references

available [1, 10, 15, 16, 30, 31, 38]. The material in this chapter will provide a rudimentary introduction to the workings of the particle filter, insomuch as it will provide a basic understanding of the particle filter implementation of the PHD-based multitarget Bayes filter summarized in Table 6.

A particle filter consists of a collection of particles with states $\{\boldsymbol{\xi}_i\}$ and corresponding weights $\{w_i\}$. They are used to approximate posterior densities, which in our case is the data-updated PHD, in the following manner:

$$\hat{D}_{k+1|k+1}(\boldsymbol{\xi}|Z^{(k+1)}) \approx \sum_{i=1}^{N} w_{i,k+1}\delta(\boldsymbol{\xi}_{i,k+1} - \boldsymbol{\xi}), \tag{97}$$

where $N$ is the number of particles and $\delta(\cdot)$ is the Dirac delta function. One can think of each particle as being a sample of the posterior density at the point $\boldsymbol{\xi}$, where the density is given by the associated weight $w_i$. The point $\boldsymbol{\xi}$ can be any point in the state space and is represented as a target state vector, consisting of target location and velocity in our multitarget tracking case. Note, however, that it is possible to have more than one particle with the same state $\boldsymbol{\xi}$, in which case the posterior density at $\boldsymbol{\xi}$ is equal to the sum of the weights of all the particles at $\boldsymbol{\xi}$.

## *3.2 The PHD Particle Filter*

Figure 2 illustrates the steps involved in particle filtering as used in the PHD-based Bayesian filter. The steps are described below, working from the top of the figure downwards and referring to the stages in the particle filter by the Roman numerals given on the lefthand-side of the figure.

### 3.2.1 Stage I: Initial Data Update

We begin in stage I with the Data Update step at time $k$. In the figure, it is assumed that we start with twelve equally-weighted particles, which one can think of as being the result of a Time Update step or of some initialization, where the weights of the particles sum to one at time $k$, since one target is expected for illustration purposes. Each particle is then weighted by the posterior density, i.e., the value of the PHD, at time $k$ to obtain $\{\boldsymbol{\xi}_i, w_i\}_k$, the set of particles and associated weights that serve as our representation of the true PHD

at time $k$. The PHD in stage I assumes that there is only one target present, and so the weights $w_{i,k}$ still sum to one.

### 3.2.2 Stage II: Resampling

Stage II is the resampling stage, in which copies of each particle $\boldsymbol{\xi}_i$ are randomly chosen with probability equal to the weight of each particle relative to the weights of the other particles. That is, each particle that survives to stage III is a copy of one of the $\boldsymbol{\xi}_i$, where the probability of selecting particle $\boldsymbol{\xi}_i$ to copy is:

$$Prob(\tilde{\boldsymbol{\xi}} = \boldsymbol{\xi}_i) = \frac{w_i}{\sum w_i}, \tag{98}$$

where $\tilde{\boldsymbol{\xi}}$ is a copy of $\boldsymbol{\xi}_i$. The weights of these new particles, $\tilde{\boldsymbol{\xi}}_i$, are equal and the total weight of the particles remains constant, i.e.,

$$\sum \tilde{w}_i = \sum w_i. \tag{99}$$

The result are the resampled particles, $\{\tilde{\boldsymbol{\xi}}_i, \tilde{w}_i\}_k$, which are passed on to stage III. This insures that particles with larger weight survive to the next stage, and particles with negligible weight die out. In such a manner, the particle filter maintains only those samples that are important in representing the density, and thus the computational complexity of the particle filter is kept to a manageable level, since the filter does not need to waste computation on particles that provide only a negligible contribution to the representation of the posterior density.

Note that the number of random samples to propagate in stage II is arbitrary. A reasonable number should be chosen to maintain good representation of the posterior density, but this also provides a mechanism for removing the increase in particles caused by the addition of the birth particles in stage III. In the example of Figure 2, even though there are twelve particles from which to choose, the resampling is performed only eight times.

Resampling is a technique used to reduce particle degeneracy, which is the unavoidable tendency of particle filters to degenerate into having all but one particle with negligible weight [1, 16]. To reduce the computational cost of having to perform resampling at each time step, one technique uses the variance of the particle weights to estimate the "effective

sample size" of the particle filter [1, 16]:

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N}(w_i)^2}, \tag{100}$$

where $N$ is the number of particles, and it provides a metric for determining when resampling should be performed. That is, when the variance of the weights grows too large, the effective sample size becomes too small, and resampling should be performed. In our simulations, resampling is performed at every time step; however, the technique described would provide a method of reducing computation time in future work.

### 3.2.3 Stage III: Time Update

Stage III in Figure 2 is the Time Update step of the PHD filter (see Table 6). The particles that survived the resampling of stage II are updated according to the target motion model. The target motion model also includes process noise, which in our simulations is Gaussian, that adds diversity to the particle states to capture any unmodeled motion, i.e., uncertainty, in the true target's behavior.[1] It is essential to have a large enough process noise to reduce the tendency towards sample impoverishment [1], which is the case in which all particles converge to a single state due to the resampling of discrete samples of the posterior, instead of from a continuous density. With too small a process noise, the posterior density may not be adequately sampled, and the particles may fail to track changes in the density. In the multitarget tracker, this could lead to a loss of target track. Caution must be taken that too large a process noise is not used, either, since too large a process noise could lead to particles being spread out too much and into areas of low importance.

To represent the birth target PHD term of the Time Update equation in Table 6, birth particles are introduced during the Time Update step in Stage III. They are represented as the four particles on the right in stage III of Figure 2. Note that, in the example, these birth particles are distributed uniformly around the area of importance (i.e., where an observation indicated that there might be a new target). As one target is expected, the sum of the birth particles is equal to one. Had they not been uniformly distributed, the

[1] This represents one possible "importance sampling density." The more general case will be discussed in Section 3.2.5.

weights of the birth particles would have been modified by (103). The number of particles in the filter has increased with the addition of the birth particles. However, as described earlier for stage II, the number of particles is reduced to the original number of particles during the resampling step.

### 3.2.4 Stage IV: Data Update Revisited

The resulting particles and weights, $\{\boldsymbol{\xi}_i, \tilde{w}_i\}_{k+1}$, are then weighted in stage IV by the posterior PHD of time $k+1$. The equation to update the weights for the PHD particle filter is given by [76, 82]:

$$w_{i,k+1} = (\sum_{n=1}^{m} u_{i,n}) \ + \ \tilde{w}_{i,k+1}(1 - p_D(\boldsymbol{\xi})) \tag{101}$$

where

$$u_{i,n} = \frac{p_D(\xi_i) f(z_n|\xi_i)\tilde{w}_{i,k+1}}{\lambda c(z_n) \ + \ \sum_{j=1}^{N} p_D(\xi_j) f(z_n|\xi_j)\tilde{w}_{j,k+1}} \tag{102}$$

for $i = 1, \ldots, N$, where $N$ is the total number of particles, $\tilde{w}_{i,k+1}$ are the weights at the end of the Time Update step (stage III), and the other variables are the parameters for the Data Update equation as described in Table 6. The result are the particles and weights $\{\boldsymbol{\xi}_i, w_i\}_{k+1}$ that represent the posterior PHD at time $k+1$. Since there are two targets present in the example at this stage, the weights sum to two.

### 3.2.5 Importance Sampling

Ideally, we would like to represent the posterior density in its entirety by using an infinite number of particles. However, greatly increasing the number of particles used in the filter eventually becomes computationally unwieldy, so we have to make do with a sub-optimal representation of the posterior density. As noted in [1], the trick is trying to find the "best" proposal distribution, also known as an importance function or importance density, from which to sample the particles, so as to most accurately represent the true posterior density while still being easy to draw samples from. This is known as importance sampling [1, 16]. Given particles that are sampled from a proposal density $q(\cdot)$, the weights of the particles are [1, 15]:

$$w_i \propto \frac{\pi(\boldsymbol{\xi}_i|z)}{q(\boldsymbol{\xi}_i|z)}, \tag{103}$$

where $\pi(\boldsymbol{\xi}_i|z)$ is the true posterior density. This is often done when it is possible to point-wise evaluate, but is hard to sample from, the true posterior density, and so a density $q(\cdot)$ that is easier to sample is used.

Importance sampling is also performed when one wants to propose particles according to some piece of information not contained in the prior, so as to achieve smarter sampling of the density. The correction weighting offsets the uneven distribution of particles introduced by the proposal density, so a uniform sampling density is maintained [30]. In the sequential importance sampling case, this correction factor relates the weights of time $k+1$ with those of time $k$:

$$w_{i,k+1} \propto w_{i,k} \cdot \frac{f(Z_{k+1}|\boldsymbol{\xi}_{i,k+1}) f(\boldsymbol{\xi}_{i,k+1}|\boldsymbol{\xi}_{i,k})}{q(\boldsymbol{\xi}_{i,k+1}|\boldsymbol{\xi}_{i,k}, Z_{k+1})}, \tag{104}$$

where $f(\boldsymbol{\xi}_{i,k+1}|\boldsymbol{\xi}_{i,k})$ is the prior likelihood, $f(Z_{k+1}|\boldsymbol{\xi}_{i,k+1})$ is the data update likelihood, and the particles at time $k+1$ are sampled from $q(\boldsymbol{\xi}_{i,k+1}|\boldsymbol{\xi}_{i,k}, Z_{k+1})$. In the case of the PHD filter, (104) is performed in two steps. During the Time Update step (see Table 6 and Section 3.2.3), the weights are updated according to:

$$\tilde{w}_{i,k+1} \propto w_{i,k} \cdot \frac{f(\boldsymbol{\xi}_{i,k+1}|\boldsymbol{\xi}_{i,k})}{q(\boldsymbol{\xi}_{i,k+1}|\boldsymbol{\xi}_{i,k}, Z_{k+1})}, \tag{105}$$

and in the Data Update step ((101) and (102)), the weights are updated according to:

$$w_{i,k+1} \propto \tilde{w}_{i,k+1} \cdot f(Z_{k+1}|\boldsymbol{\xi}_{i,k+1}). \tag{106}$$

Note that the states of the particles $\boldsymbol{\xi}_i$ are determined for time $k+1$ during the Time Update step, so it is there that the correction factor is applied when the particles propagated from time $k$, and the birth particles for time $k+1$, are distributed according to anything other than the prior likelihood. For example, if the particles propagating from time $k$ to $k+1$ were not sampled in the Time Update step according to the prior, but according to a density that incorporated knowledge of the observations at time $k+1$, then the particles would already be distributed according to some knowledge of $f(Z_{k+1}|\boldsymbol{\xi}_{i,k+1})$. However, in the Data Update step, the $f(Z_{k+1}|\boldsymbol{\xi}_{i,k+1})$ weighting of (106) assumes that the particle weighting and sampling distribution only contains knowledge of the prior. Hence, the correction factor is needed, so that the final particle weights and distribution for time $k+1$

are not doubly affected by the observations, which would cause the particle filter to have an incorrect representation of the posterior density.

The correct particle weights for the propagated particles and birth particles at the end of the Time Update step in the PHD particle filter are given by Vo *et al.* in [76]. For the propagated particles:

$$\tilde{w}_{i,k+1} = \frac{s_{k+1|k}(\boldsymbol{\xi}_{i,k}) f_{k+1|k}(\boldsymbol{\xi}_{i,k+1}|\boldsymbol{\xi}_{i,k}) + \hat{b}_{k+1|k}(\boldsymbol{\xi}_{i,k+1}|\boldsymbol{\xi}_{i,k})}{q_{k+1}(\boldsymbol{\xi}_{i,k+1}|\boldsymbol{\xi}_{i,k}, Z_{k+1})}, \tag{107}$$

where the propagated particles are sampled from the proposal density $q_{k+1}(\boldsymbol{\xi}_{i,k+1}|\boldsymbol{\xi}_{i,k}, Z_{k+1})$, and the other parameters are as specified in Table 6.

For the birth particles:

$$\tilde{w}^{birth}_{i,k+1} = \frac{\hat{b}_{k+1|k}(\boldsymbol{\xi}_{birth,i})}{J_{k+1} \cdot q_{k+1}(\boldsymbol{\xi}_{birth,i}|Z_{k+1})}, \tag{108}$$

where $J_{k+1}$ is the number of birth particles, which are sampled from the proposal density $q_{k+1}(\boldsymbol{\xi}_{birth,i}|Z_{k+1})$, and $\hat{b}_{k+1|k}(\boldsymbol{\xi}_{birth,i})$ is the PHD of the spontaneous birth targets, as described in Table 6 and Section 2.2.4.1. An example of this weighting is found in Section 4.3.3.

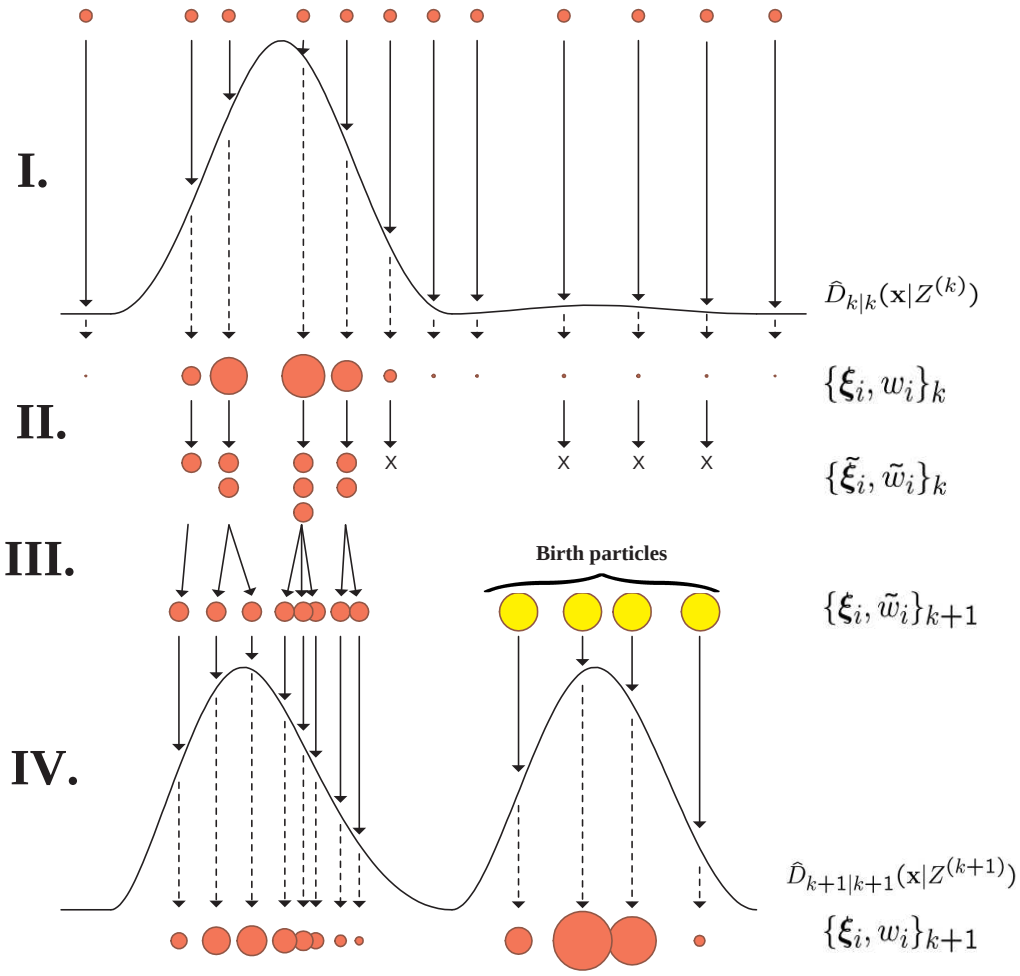



**Figure 2:** The stages in the PHD particle filter. The particles are represented by the circles, and the size of the circles indicates the weight associated with each particle. **Stage I**: the Data Update step performed on an initial distribution of particles. One target is present. The resulting particle weights sum to one. **Stage II**: the resampling stage. **Stage III**: the Time Update step. The weights of the propagated particles still sum to one. The weights of the birth particles also sum to one. **Stage IV**: the Data Update step. Two targets are present, and the resulting particle weights sum to two. This figure is adapted from [15].

# CHAPTER IV

# MULTITARGET, MULTISENSOR TRACKER: INITIAL IMPLEMENTATION

## *4.1 Introduction*

Having introduced finite-set statistics (FISST) and the probability hypothesis density (PHD) in Chapter 2 and the concept of a particle filter for the PHD-based multitarget Bayes filter equations in Chapter 3, we now present our initial application of the PHD-based particle filter to a realistic multitarget, multisensor passive radar scenario. However, before presenting our implementation and its results, a brief introduction to passive coherent location[1] is worthwhile.

## *4.2 Passive Coherent Location (PCL)*

### 4.2.1 Range

Consider a bistatic radar consisting of a passive receiver and an independent transmitting antenna. If the direct path signal is measured along with the reflected path signal, then correlation processing yields the following range measurement observation:

$$R = \sqrt{(x - x_t)^2 + (y - y_t)^2} + \sqrt{(x - x_r)^2 + (y - y_r)^2}, \tag{109}$$

where $(x_r, y_r)$ and $(x_t, y_t)$ are the locations of the antennas, and $(x, y)$ is the location of the target. Thus, a target can be located along an ellipse, where the receiver and transmitter are located at the foci of the ellipse, as seen in Figure 3.

It is difficult to build highly directional receiver antennas that operate at the low frequencies of interest in a passive radar system that exploits FM broadcasts. Hence, rather

[1] To the best of our knowledge, the term "passive coherent location" was coined by Dick Lodwig of Lockheed Martin (then, IBM) and his colleagues.

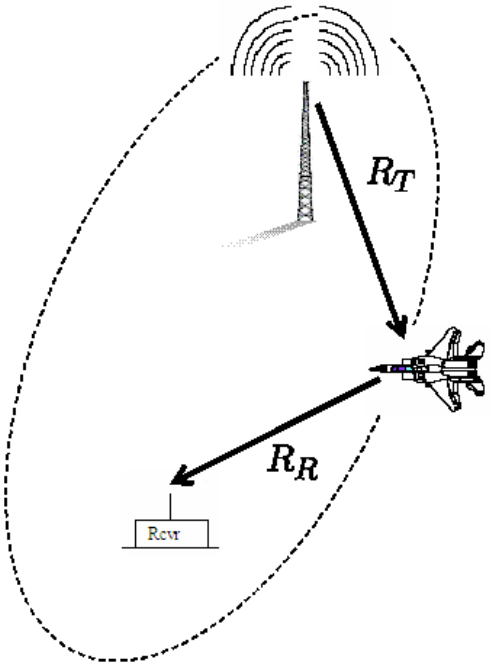



**Figure 3:** Bistatic range ellipse generated by a target and a receiver-transmitter pair. The observed range measurement $R = R_T + R_R$. (Figure not to scale.)

than exploit angle-of-arrival information to resolve a target's location, multiple transmitter-receiver pairs are usually employed instead. The target can thus be located at the intersection of the resulting bistatic range ellipses. This is not to imply that angle information, if available in a PCL system, is of no value; we simply wish to explore the limits of what can be achieved without it. The effect of including angle information is studied in Section 6.5.

A problem that arises, however, is that of ghost targets. A ghost target appears at the intersection of bistatic range ellipses where no target is present. This is due to the nature of the ellipse geometry (see Figure 4) and confuses multitarget trackers, which must process ghost targets until they disappear. Noisy measurements exacerbate the problem. The PHD-based particle filter, however, will be seen to adequately handle ghost targets with no additional conceptual effort, i.e., no explicit ghost-busting[2] logic is needed.

[2]See the following website for a description of a ghost-busting technique: http://www.clw.org/nmd/businessweek.html

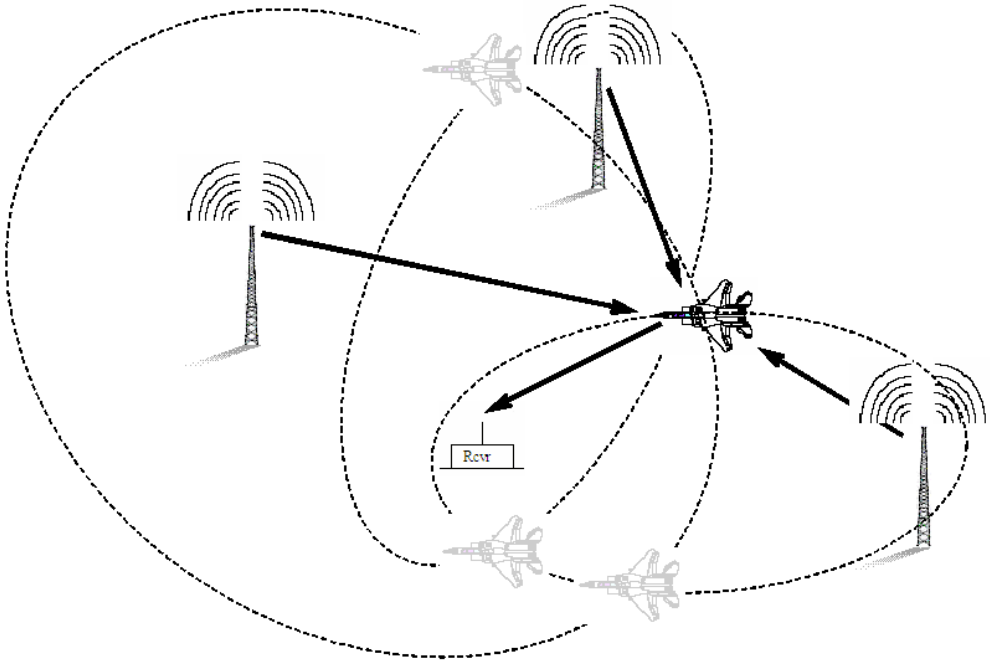



**Figure 4:** PCL of a target using three receiver-transmitter pairs. The true target is in black; the rest represent ghost targets. (Figure not to scale.)

### 4.2.2 Velocity

By observing the Doppler shift caused by the target in the received signal's frequency, a bistatic radar also provides the rate of change of the range measurement given in (109):

$$\dot{R} = \frac{(x - x_r)\dot{x} + (y - y_r)\dot{y}}{\sqrt{(x - x_r)^2 + (y - y_r)^2}} + \frac{(x - x_t)\dot{x} + (y - y_t)\dot{y}}{\sqrt{(x - x_t)^2 + (y - y_t)^2}}. \tag{110}$$

With $\dot{R}$ measurements from multiple transmitter-receiver pairs, a target's velocity components $(\dot{x}, \dot{y})$ can be found.

## *4.3 Scenario Configuration*

The PHD particle filter is applied to a tracking problem in a realistic PCL scenario. The Field of View (FoV) consists of an 80 km × 80 km stretch of the Washington D.C. area. The receiving antenna is in the middle of the FoV and is assumed to be located on one of the Lockheed Martin Mission Systems buildings. A receiver such as Lockheed Martin's *Silent Sentry*® system, except with a simpler antenna, is assumed. The illuminators of

opportunity consist of three non-cooperative FM transmitters. The transmitter specifications are given in Table 7, and their locations can be seen in Figure 5(a). The receiver coordinates and system specifications are listed in Table 8. All antennas are assumed to be omni-directional, and thus they have unity gain. The noise figure listed, which is meant to account for external interference sources as well as internal receiver noise, is assumed to be a valid approximation for an urban environment such as Washington D.C. [17].

**Table 7:** Transmitting Antenna Specifications in the Washington, D.C. Simulation

| Call Letters | Latitude | Longitude | Frequency ($f$) | Power ($P_T$) | Bandwidth ($\beta$) |
|---|---|---|---|---|---|
| WAMU | 38.936 °N | 77.093 °W | 88.5 MHz | 50.0 kW | 45 kHz |
| WETA | 38.892 °N | 77.132 °W | 90.9 MHz | 75.0 kW | 45 kHz |
| WPGC | 38.864 °N | 76.911 °W | 95.5 MHz | 50.0 kW | 45 kHz |

**Table 8:** Receiver System Specifications in the Washington, D.C. Simulation

| | |
|---|---|
| Latitude | 39.153 °N |
| Longitude | 77.215 °W |
| Coherent Processing Interval ($CPI$) | 0.5 sec |
| Reference Temperature ($T_0$) | 290 K |
| Noise Figure ($NF$) | 30 dB |
| Gain ($G_R$) | 0 dB |

### 4.3.1 The PHD-Based Particle Filter

As mentioned at the conclusion of Section 2.2.4.3, the PHD filter intrinsically handles the calculation of the expected number of targets in the region under observation, and it locates targets without performing explicit report-to-track association. We thus expect the PHD to be an especially useful tool for handling the many ghost targets that arise from noisy bistatic radar measurements in PCL. Furthermore, since the PHD filter provides a painless way to fuse together multiple sensor data, we compare the tracking results of the PHD particle filter using both range and Doppler observations to the tracking results when range-only observations are used.

We use the particle filter implementation of the Bayesian PHD update equations, described in Chapter 3, whereby the PHD is represented by a collection of particles and their corresponding weights. At time-step $k$, each particle in the filter is a vector of the form $\xi_i = [x_i \ y_i \ \dot{x}_i \ \dot{y}_i]^T$ and has a weight $w_{i,k}$, where $(x_i, y_i)$ specify the particle's location and $(\dot{x}_i, \dot{y}_i)$ specify its velocity components. As per the defining property of the PHD, the estimate of the number of targets present is

$$\tilde{N} = E[no.\ of\ targets] = [N_{k|k}]_{nearest\ integer}, \tag{111}$$

where

$$N_{k|k} = \sum_i w_{i,k}. \tag{112}$$

### 4.3.2 Initialization

The simulation begins by independently and randomly assigning the particles' $x$ and $y$ components to fall within the FoV. The $\dot{x}$ and $\dot{y}$ components are independently and randomly chosen to be between a minimum of $-495$ km/h and a maximum of 495 km/h (i.e., $-137.5$ m/s to 137.5 m/s), where North and East are positive. The particle weights are initialized to zero, since we do not expect any targets to be present at time $k = 0$.

### 4.3.3 Time Update

The time-update step of the particle filter involves multiplying each particle vector by a simple constant-velocity transition matrix and adding Gaussian process noise $u$:

$$\boldsymbol{\xi}_{k+1|k} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \boldsymbol{\xi}_k + \begin{bmatrix} \frac{T^2}{2} \\ \frac{T^2}{2} \\ T \\ T \end{bmatrix} \cdot u, \tag{113}$$

where $T$ is the duration of the time step, which in this case is one second. This propagates the particles forward in time, thus modeling the target motion, where each time step $k$ of simulation represents one second of time.

To model the PHD of new targets that appear in the FoV, birth particles are added to the simulation during the time-update step. They indicate where new targets are likely

to appear at the current time step. To economize on the number of particles needed in the simulation and to achieve better target tracking results, we propose a targeted cluster placement of birth particles to be used wherever a bistatic range ellipse intersects with the edge of the FoV.[3] At the location of the intersection, a cluster of birth particles is centered and spread independently in both $x$ and $y$ according to a normal distribution with a standard deviation equal to a bistatic range cell (see Section 4.4.5 and Table 9). When no bistatic range ellipses intersect the FoV boundaries, the birth particles are randomly placed uniformly in a 9 km-wide band around the inside edge of the FoV.

In both placement methods, the velocity components of the particles are initialized independently and randomly with uniform probability over all possible velocities, as given in Section 4.3.2. However, if a particle is placed in the right-hand quadrant of the FoV, then its initial $\dot{x}$ component is restricted to negative values. If it is placed in the left-hand quadrant, then the $\dot{x}$ component is initialized to positive values only. A similar restriction is enforced on the initial $\dot{y}$ component of a particle that is placed in either the top or bottom quadrants of the FoV.

The simulation assumes that targets will not spontaneously disappear and that they will not spawn new targets. Any particles, whose $x$ and $y$ components place them outside of the FoV, have their location components adjusted, so that they are repositioned in a mirror-image fashion across the nearest FoV edge back into the FoV. This keeps all of the particles inside the region of interest.

We now weight the particles according to the method described in [76] and Section 3.2.5 for particle filter representations of the PHD. Since we simply use the prior target-motion model to propagate the particles from the previous time step, these propagated particles maintain the same weights as they had at the end of the previous time step. The birth particles, when the uniform placement method is used, are given equal weighting. When the targeted cluster placement is used, however, the birth particles are given weights

$$\tilde{w}_{birth_{i,k+1}} = \frac{1}{J_{k+1} \cdot Q_x \cdot q_x(x_{i,k+1}|z_{k+1}) \cdot Q_y \cdot q_y(y_{i,k+1}|z_{k+1})}, \tag{114}$$

[3] An alternate technique will be presented in Section 5.3.

where $J_{k+1}$ is the number of birth particles used and $q_x(x_{i,k+1}|z_{k+1})$ and $q_y(y_{i,k+1}|z_{k+1})$ are normal density functions with means equal, respectively, to the $x$ and $y$ positions of the ellipse intersection at the FoV edge and with standard deviations equal to the intersection's corresponding bistatic range cell. $Q_x$ is set to 2, if the intersection occurs along the left or right edge of the FoV, and is set to 1, otherwise. Similarly, if the intersection occurs along the top or bottom edges, then $Q_y$ is set to 2; otherwise, it is set to 1. This takes into account the doubling of particle density due to the folding-in of particles found outside the FoV.

In both birth particle placement methods, we normalize the birth particle weights, such that $\sum \tilde{w}_{birth_{i,k+1}}$ equals the expected number of new targets per scan. Since we assume that only one target might enter the FoV at each time step, we set this term equal to one. However, one could choose a higher or lower value if an alternative birth model is desired. This step is not explicitly mentioned in [76], but we found this normalization necessary to have the particle filter accurately represent the PHD.

The results of the time-update step are the propagated and birth particles and their associated weights, indicated by $\tilde{w}_{i,k+1}$, which represent the predicted PHD for time-step $k+1$.

### 4.3.4 Data Update

In the data-update step, the time-predicted $\tilde{w}_{i,k+1}$ are converted to the final PHD particle weights, $w_{i,k+1}$, by incorporating the radar range and Doppler observations at time $k+1$. Given a single sensor with the set of observations $Z_s = \{z_1, \ldots, z_m\}$ made at time $k+1$, probability of detection $p_D(\boldsymbol{\xi})$, single-target likelihood function $f(\mathbf{z}|\boldsymbol{\xi})$ and Poisson-distributed false alarms with parameter $\lambda$ and density $c(\mathbf{z})$, the data-updated weights are computed by (see Sections 2.2.4.2 and 3.2.4):

$$w_{i,k+1} = (\sum_{n=1}^{m} u_{i,n}) \ + \ \tilde{w}_{i,k+1}(1 - p_D(\boldsymbol{\xi})), \tag{115}$$

where

$$u_{i,n} = \frac{p_D(\xi_i) f(z_n|\xi_i) \tilde{w}_{i,k+1}}{\lambda c(z_n) \ + \ \sum_{j=1}^{N} p_D(\xi_j) f(z_n|\xi_j) \tilde{w}_{j,k+1}}, \tag{116}$$

for $i = 1, \ldots, N$, where $N$ is the total number of particles.

The set of observations $Z_s$ contains both range and Doppler measurements. Thus, either $f_R(z_n|\xi_i)$ or $f_{\dot{R}}(z_n|\xi_i)$ must be used as the single-target likelihood function $f(z_n|\xi_i)$, depending on whether $z_n$ is a range or a Doppler observation, respectively. The computations of $p_D(\boldsymbol{\xi})$, $f(\mathbf{z}|\boldsymbol{\xi})$, $\lambda$ and $c(\mathbf{z})$ are given in Section 4.4.

In the bistatic radar case, each receiver and transmitter pair constitutes a "sensor." In our example, there are three sensors in the configuration, and three sets of range and Doppler observations are collected at each time step, namely $\{Z_1, Z_2, Z_3\}$. Following a procedure suggested by Mahler [56] to determine the final weights for this multisensor case, (115) and (116) are first applied to $Z_1$. The resulting $w_{i,k+1}$ are then used as the $\tilde{w}_{i,k+1}$ to reiterate (115) and (116) over $Z_2$. The latter procedure is repeated for $Z_3$ to find the final multisensor particle weights. The order in which the observation sets are processed does affect the final result; although, practically, it has little effect. This is thought to be due to the Poisson-approximation of the predicted PHD that is made to derive the Data Update equation for the posterior PHD (see Section 2.2.4.2). Because we attribute these variations in the result to the approximation made in the Data Update step, we did not deem it necessary to pursue this issue further. However, finding a solution to this problem, such as formulating an alternative to the approximation in the derivation of the Data Update equation, remains available for further investigation. Mahler suggested tackling this problem by combining the sensors into a single "pseudo-sensor" approximation [48]. However, our simulations found the "pseudo-sensor" approximation to be an incorrect solution, since it caused single targets to generate multiple observations. Via e-mail correspondence[4], Mahler acknowledged that the "pseudo-sensor" approximation was incorrect, because it violates the measurement model used to derive the Data Update equation. He suggested either re-deriving the PHD filter using a measurement model that allows single targets to generate multiple observations (although Mahler claimed that this appeared to be computationally intractable, it remains open for further investigation) or using the technique described above of sequentially processing the sensor observations.

[4]E-mail correspondence between June 16-25, 2003.

Having generated the final particle weights, $w_{i,k+1}$, the expected number of targets in the FoV is computed via (96). The locations of the $\tilde{N}$ expected targets are found by extracting the $\tilde{N}$ highest peaks from the PHD represented by these weights. An expectation-maximization algorithm was used for this extraction. An alternate method is considered in Section 5.2.4.

### 4.3.5 Peak Extraction

To find the target locations and their velocities, the $\tilde{N}$ highest peaks must be extracted from the PHD. To find these peaks, we assume that the PHD in the neighborhood of the peaks can be approximated by Gaussian distributions, so we attempt to fit a mixture of Gaussians to the PHD using an expectation-maximization (EM) algorithm [5], which we modify to account for the particle weights. Thus, the algorithm to find $\theta_g = (\alpha_g, \mu_g, \Sigma_g)$, which are the weight, mean and covariance parameters of the $g$-th Gaussian distribution in the mix, is given by the following iteration:

Expectation:

$$P(g|x_i) = \frac{p(x_i|\theta_g)\alpha_g}{\sum_{j=1}^{G} p(x_i|\theta_j)\alpha_j}, \tag{117}$$

Maximization:

$$\alpha_g^{new} = \sum_{i=1}^{N} w_i P(g|x_i), \tag{118}$$

$$\mu_g^{new} = \frac{1}{\alpha_g^{new}} \sum_{i=1}^{N} w_i P(g|x_i) x_i,$$

$$\Sigma_g^{new} = \frac{1}{\alpha_g^{new}} \sum_{i=1}^{N} w_i P(g|x_i)(x_i - \mu_g^{new})(x_i - \mu_g^{new})^T,$$

where $p(x_i|\theta_j)$ is a normal density function with mean $\mu_j$ and covariance matrix $\Sigma_j$, $G$ is the number of Gaussians in the mixture and $G \geq \tilde{N}$, where $\tilde{N}$ is the number of targets estimated by integrating the PHD via Eq. (96).

The $\mu_g$ are initialized by randomly choosing $G$ particles and selecting their components to be the values for the $\mu_g$. To obtain good results from the EM algorithm [4], short runs of the algorithm are performed, and the run that produces the highest likelihood is then used

for a longer EM run. The result is the final estimate of the $G$-Gaussian mixture. When iterating the EM algorithm, a run is terminated upon achieving a given threshold or if a covariance matrix becomes singular. The preceding is performed multiple times for different values of $G$, and a minimum description length (MDL) criterion is then used to select the best fitting Gaussian mixture by maximizing the penalized likelihood [20]:

$$L(\mathbf{x};\theta_G) - \frac{\rho}{2}\ln N, \tag{119}$$

where $\rho = (G-1) + G\left(d + \frac{d(d+1)}{2}\right)$ and $d$ is the particle dimensionality (in the current case, $d = 4$), and where

$$L(\mathbf{x};\theta_G) = \frac{N}{\sum w_i}\sum_{i=1}^{N} w_i \ln \sum_{j=1}^{G} \alpha_j p(x_i|\theta_j). \tag{120}$$

The means of the $\tilde{N}$ highest-weighted Gaussians in the best fitting mixture are then taken to be the expected locations and velocities of the targets. A benefit of using the EM algorithm is that it produces covariance matrices that provide one with a measure of uncertainty in the location and velocity estimates.

### 4.3.6 Resampling

Before iterating the particle filter over the next time step, the particles are resampled via a Monte Carlo method, as described in Section 3.2.2, to obtain an initial number (i.e., the amount before birth particles were added) of equally weighted particles, where

$$\sum_i w_i = \sum_i w_{i,k+1}. \tag{121}$$

## *4.4 Bistatic Radar Variables*

### 4.4.1 Signal-to-Noise Ratio, SNR

To compute $f(\mathbf{z}|\boldsymbol{\xi})$ and $p_D(\boldsymbol{\xi})$, it is first necessary to compute each sensor's signal-to-noise ratio for each particle. The SNR of a particle is calculated as follows [26, 27, 80]:

$$\mathrm{SNR}(\xi_i) = \frac{K}{R_T^2 R_R^2}, \tag{122}$$

where $R_T$ and $R_R$ are the distances between the particle's $(x, y)$ location and the sensor's transmitting and receiving antennas, respectively, and

$$K = \frac{P_T\, G_T\, G_R\, \lambda_f^2\, \sigma_{rcs}\, F_T^2 F_R^2}{(4\pi)^3 k\, T_0 (\frac{1}{CPI})(NF)}, \tag{123}$$

where $\lambda_f = \frac{c}{f}$, $c$ is the speed of light, and $f$ is the frequency of the FM signal given in Table 7. The transmitter power $P_T$ is also taken from Table 7, and the transmitter gain $G_T$ is assumed to be unity. The receiver gain $G_R$, reference temperature $T_0$, coherent processing interval $CPI$ and noise figure $NF$ are taken from Table 8. Boltzmann's constant is represented by $k$, and $F_T$ and $F_R$ are the signal propagation factors. For this initial study, it was assumed that signal propagation gains and losses are negligible; including such effects is studied in Section 6.4. The target's bistatic radar cross section is denoted by $\sigma_{rcs}$. All targets in the simulation are assumed to have $\sigma_{rcs} = 10$ dBsm.

### 4.4.2 Probability of Detection, $p_D$

The calculation of the bistatic radar's probability of detection is based on its SNR and the probability of false alarm, $p_{FA}$. At low frequencies, a target may reasonably be assumed to be slowly fluctuating; hence, a Rician target model is employed. Thus, [39]

$$p_D(\boldsymbol{\xi}) = Q\left[\sqrt{2\,\mathrm{SNR}(\boldsymbol{\xi})}\,, \sqrt{2\ln\left(\frac{1}{p_{FA}}\right)}\,\right], \tag{124}$$

where $Q$ is the Marcum Q-function, $\mathrm{SNR}(\boldsymbol{\xi})$ is given by Eq. (122), and $p_{FA}$ is set to a fixed value. For a fixed $p_{FA}$, a gain in SNR corresponds to an increase in $p_D$.

In the simulation, the $p_{FA}$ is initially set to $10^{-4}$. This achieves a $p_D = 0.9999$ with an SNR = 14.94 dB, and a $p_D = 0.1$ when SNR = 6.19 dB [39]. For reasonable simulation, $p_D$ is restricted to a maximum value of 0.99999.

Note that the $p_D(\boldsymbol{\xi})$ in (101) does not depend on any specific radar observation, since the $(1 - p_D(\boldsymbol{\xi}))$ term deals with potential missed targets. Thus, a $\sigma_{rcs}$ must be chosen that one would expect a potential missed target to have were the radar to detect it. For illustration, we escape this vexing chicken-and-egg situation by choosing $\sigma_{rcs} = 10$ dBsm, since this is the value assumed in generating the simulated data.

### 4.4.3 Single-Target Likelihood, $f(\mathbf{z}|\boldsymbol{\xi})$

#### *4.4.3.1 Range Likelihood, $f_R(\mathbf{z}|\boldsymbol{\xi})$*

The single-target range likelihood function of each bistatic radar antenna pair determines how close each particle's $(x, y)$ values are to the observed target location, given that the radar observes the range measurement given by (109). Each particle's corresponding bistatic range measurement is computed ($R_{\xi_i}$), as well as the difference between it and the observed range.

$f_R(z_i|\xi_i)$ is a normal density function with mean $R_{\xi_i}$ and variance $\sigma_r^2$, where $\sigma_r^2$ is the variance of the bistatic range:

$$\sigma_r^2 = \sigma_t^2 \cdot c^2 \tag{125}$$

and [3]

$$\sigma_t^2 = \frac{1}{2\,\beta^2\,\mathrm{SNR}(\xi_i)}, \tag{126}$$

where $\beta$ is the transmitter bandwidth specified in Table 7, and $\mathrm{SNR}(\xi_i)$ is given by (122).

#### *4.4.3.2 Doppler Likelihood, $f_{\dot{R}}(\mathbf{z}|\boldsymbol{\xi})$*

The single-target Doppler likelihood function of each bistatic radar pair determines how close the $\dot{R}$ value of each particle, given by substituting the particle's components into (110), is to the observed $\dot{R}$ measurement. Each particle's corresponding rate of bistatic range change is computed ($\dot{R}_{\xi_i}$), as well as the difference between it and the observed $\dot{R}$.

$f_{\dot{R}}(z_i|\xi_i)$ is a normal density function with mean $\dot{R}_{\xi_i}$ and variance $\sigma_{\dot{r}}^2$, where $\sigma_{\dot{r}}^2$ is the variance of the rate of change in bistatic range:

$$\sigma_{\dot{r}}^2 = \sigma_f^2 \cdot \lambda_f^2, \tag{127}$$

where $\lambda_f$ is the frequency of the transmitted signal, and [26, 27]

$$\sigma_f^2 = \max\left[\frac{3}{2\,\mathrm{SNR}\cdot(\pi\cdot CPI)^2}\ ,\ \frac{1}{CPI^2}\right]. \tag{128}$$

On the right-hand side of (128), the first term in the max function is the accuracy with which the bistatic radar is able to measure the received signal. The second term is the resolution obtained from the passive radar's use of the discrete Fourier transform to compute the

Doppler shift of the signal. Thus, the variance in the $\dot{R}$ measurement is the worse, i.e. greater, of the two terms.

### 4.4.4 False Alarm Parameters, $\lambda$ and $c(\mathbf{z})$

The false alarm Poisson-distribution parameter $\lambda$ is computed based on the number of range and Doppler cells present in the simulation. These, in turn, are based on the extent of range and Doppler in the scenario, as well as the bistatic range and Doppler resolutions of the radar.

Each transmitter-receiver pair's $\lambda$ parameter is calculated in the following manner:

$$\lambda = (total\ no.\ cells) \times p_{FA}, \tag{129}$$

where

$$total\ no.\ cells = (no.\ range\ cells) \times (no.\ Doppler\ cells), \tag{130}$$

and

$$no.\ range\ cells = \frac{range\ extent}{bistatic\ range\ resolution}, \tag{131}$$

$$no.\ Doppler\ cells = \frac{Doppler\ extent}{Doppler\ resolution}, \tag{132}$$

and

$$bistatic\ range\ resolution = \frac{c}{\beta}, \tag{133}$$

$$Doppler\ resolution = \frac{1}{CPI}, \tag{134}$$

and

$$range\ extent = 1.5 \times \sqrt{(80\text{km})^2 + (80\text{km})^2}, \tag{135}$$

$$Doppler\ extent = 2\left(\frac{2V_{max}}{\lambda_f}\right), \tag{136}$$

where $V_{max}$ is the maximum possible target velocity. The range extent value of (135) is for the hypothetical case where the receiver is located at the center of the FoV, the transmitter is located in a corner of the FoV, and the target is located at the opposite corner. The Doppler extent found in (136) takes into account both positive and negative velocities. Thus, both extents are chosen to be as large as theoretically possible in our scenario.

The false alarms are assumed to be uniformly distributed over the range and Doppler extents, and thus the spatial distribution parameter is determined in the following manner:

$$c(\mathbf{z}) = \frac{1}{(range\ extent) \times (Doppler\ extent)}. \tag{137}$$

Note that

$$p_{FA} = \lambda \cdot c(\mathbf{z}) \cdot (range\ resolution) \cdot (Doppler\ resolution). \tag{138}$$

### 4.4.5 Bistatic Range Cells

To place the birth particles correctly in the targeted clustering method described in Section 4.3.3, the size of the bistatic range cell at the cluster's location is computed. A bistatic range cell is the resolution at which a bistatic radar can pinpoint a target's location. It is approximated by [80]:

$$\Delta R_B \approx \frac{c\tau}{2\cos(\frac{\psi}{2})}, \tag{139}$$

where $\tau$ is the compressed pulse width and $\psi$ is the bistatic angle:

$$\psi = \cos^{-1}\left[\frac{R_T^2 + R_R^2 - L^2}{2R_R R_T}\right], \tag{140}$$

where $L$ is the distance between the transmitter and the receiver. Thus,[5]

$$\begin{aligned} \Delta R_B &\approx \frac{c\tau\sqrt{R_R R_T}}{\sqrt{(R_T + R_R)^2 - L^2}} \\ &= \frac{c\sqrt{R_R R_T}}{\beta\sqrt{(R_T + R_R)^2 - L^2}}. \end{aligned} \tag{141}$$

## *4.5 Simulation*

The simulation contains two targets. The first enters at time $k = 7$ at location (80 km, 20 km) on the East edge of the FoV and travels at $-395$ km/h (i.e., 109.7 m/s West). The second enters at time $k = 9$ from location (50 km, 0 km) on the South edge and travels North at 340 km/h (94.4 m/s). The process noise in the time-update step is assumed to have a standard deviation of 5 m/s. An example of the PHD particle filter is given in Figure 5.

[5]This is a correction to the derivation found in [72].

### 4.5.1 Typical Measurements

Table 9 lists approximate ranges of values observed for the variables introduced in Sec. 4.4.

**Table 9:** Approximate Value Ranges of Bistatic Radar Variables in the Washington, D.C. Simulation

| Variable | typ. min | typ. max |
|---|---|---|
| SNR | 12.2 dB | 32.5 dB |
| $p_D$ | 0.9418 | 0.99999 |
| $\sigma_r$ | 3.94 m | 1.42 km |
| $\sigma_{\dot{r}}$ | 6.278 m/s | 6.775 m/s |
| $\lambda$ | 0.213 | 0.229 |
| $c(z)$ | $3.4 \times 10^{-8}$ | $3.6 \times 10^{-8}$ |
| $\Delta R_B$ | 3.33 km | 4.31 km |

The false alarm parameters result in roughly two false alarms for every three time step iterations of the PHD filter. Within a single time step, the number of false alarms typically ranges from zero to four. Since the $\frac{1}{CPI^2}$ term in (128) dominates in the current scenario, the value of $\sigma_{\dot{r}}$ depends only on the three transmitters being used. Thus, it does not acquire a wide range of values in the simulation.

### 4.5.2 Results with $p_{FA} = 10^{-4}$

**Target Tracking** We first tried the range and velocity tracker using 1,000 particles and an additional 500 birth particles. However, as seen in Figure 6, the range and velocity tracker did not show a drastic improvement over the range-only tracker. Also, the range and velocity tracker would occasionally fail to detect the second target; whereas, the range-only tracker appeared always to detect both targets.

In an attempt to improve the range and velocity tracker, we doubled the number of particles to 3,000 (i.e., 2,000 propagated and 1,000 birth particles). The tracking results are displayed in Figure 7. The tracking performance of the range-only filter, when used with 3,000 particles, is shown in Figure 8. A close-up of the two are depicted in Figure 9. One can see that, with the additional particles, the range and velocity tracker locates the

targets with a tighter track than does the range-only tracker, especially when the targets are located far apart from each other. In Figure 9, it appears that there is a slight bias in the range and velocity tracker's results in locating the first target. However, the results are still within a bistatic range cell, and the bias did not appear on subsequent simulation runs. Figure 10 displays the estimated target velocity values found by the range and velocity filter; they appear to track the actual target velocities.

The 3,000 particle range and velocity tracker detected both targets. However, unlike the range-only tracker, which detected both targets immediately, the range and velocity tracker immediately detected the first target but took a few iterations (latencies of 1 to 18 time steps have been observed) to detect the second target. This should not summarily be considered a flaw in the PHD filter, since we should not expect the tracker to detect a target instantly upon its entry into the FoV. Rather, we would expect a brief data accrual time during which the tracker filters out ghost target ambiguity. This may also, however, be due to a lack of a sufficient number of particles being used.

**Target Number Estimation** In terms of detecting the correct number of targets at each time step, the range-only tracker, at first glance, appeared to perform slightly better. For example, it overestimated by one the number of targets in twenty time steps; whereas, the range and velocity tracker once underestimated by two, once overestimated by one, and twenty times underestimated by one. However, the underestimation errors of the range and velocity filter occurred at the beginning of the simulation run during which time the filter was trying to detect both targets; whereas, the overestimation errors in the range-only filter occurred throughout the simulation. Thus, the range and velocity tracker demonstrated more stability in maintaining a correct estimate of the number of targets present.

**Peak Extraction** It must also be pointed out, as seen in Figures 6 and 9, that even though the PHD particle filter correctly tracked the two targets in both the 3,000 particle range and velocity tracker and the range-only tracker, it did not correctly extract the target locations at every time step. This was due to the behavior of the peak-extraction algorithm, which occasionally produced singular covariance matrices or fit two Gaussians

to the same peak, thereby locating the same target twice; whereas, a visual inspection of the PHD clearly showed two targets. Increasing the number of particles did ameliorate the peak extraction, since there were more data points available to which to fit the Gaussians. The development of improved peak-extraction algorithms is an important area of PHD research in general; some steps in this direction are suggested in [37], and we introduce a peak-extraction technique that exploits the properties of the PHD in Section 5.2.4.

### 4.5.3 Results with $p_{FA} = 10^{-2}$

The 3,000 particle range-only and range and velocity simulations were performed once more but with $p_{FA} = 10^{-2}$. Since the SNR in the simulations did not change, this increase in probability of false alarm resulted in an increase in $p_D$, as given by (124). The typical minimum probability of detection, as given in Table 9, increased to 0.9978. Within a single time step, the average number of false alarms was 67.

As seen in Figure 11, the range-only tracker could not track the two targets. Furthermore, it was completely unable to estimate the correct number of targets present. At any given time step, it estimated there being between 5 and 31 targets present. On the contrary, the range and velocity tracker was able to track both targets, as shown in Figures 12 and 13. It overestimated the number of targets about 10%-20% of the time (the maximum by which it overestimated was four) and underestimated about 2.5% of the time.

The range and velocity filter with $p_{FA} = 10^{-2}$ did occasionally exhibit behavior similar to that of the 1,500 particle range and velocity tracker described in Section 4.5.2. That is, it only tracked the first target, or it tracked the first target for a brief period before dropping it and tracking only the second target for the remainder of the simulation. Other times, it tracked both targets as it should.

## *4.6 Conclusions*

### 4.6.1 Importance of High SNR

Initially, the range-only simulation was run using a noise figure of 45 dB to explore a worst-case scenario. However, this was found to be an inappropriate model of the Washington D.C. scenario, since it produced an SNR that was too low over most of the 80 km $\times$ 80 km

coverage area. With such poor SNR, the $p_D$ was only above 0.9 for the immediate area around the antenna pairs, while most of the FoV had a $p_D$ close to zero.

The prevalence of low $p_D$ caused the filter to considerably overestimate the number of targets present. Because the birth particles that were added to the filter in the time-update step were located in an area of low $p_D$, the filter predicted many targets that did not exist. This is logical, since the filter can only assume that targets are appearing in the area of low SNR based on the birth particle model. It does not receive any radar observation information to contradict the assumption.

We conjectured that restricting birth particle placement to regions of high SNR, or simply restricting the FoV to include only regions of sufficiently high SNR, will mitigate the effect of areas of low $p_D$ on the filter. This matter is addressed again in Sections 5.2.7 and 6.3.

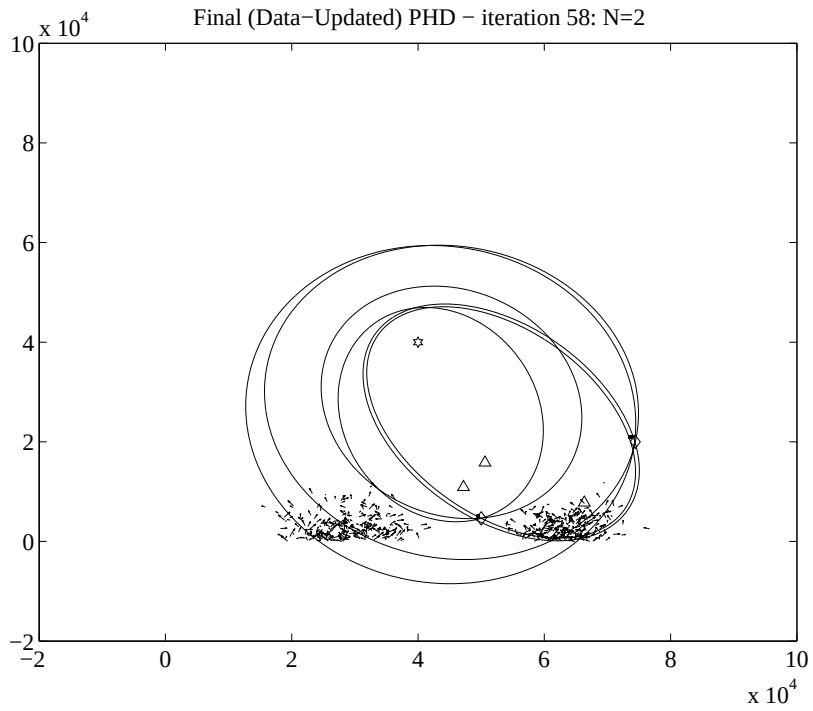
### 4.6.2 The PHD Particle Filter

Incorporating Doppler measurements into the PHD-based particle filter was expected to improve the tracking ability of the filter, since the filter would possess additional information about the targets being tracked. The introduction of Doppler measurements, however, while providing additional velocity information, also enlarged the discrepancy between the proposal density used by the particle filter and the true posterior PHD. In the $p_{FA} = 10^{-4}$ case, to achieve at least an equivalent tracking and ghost-busting performance as that of the range-only filter, the number of particles had to be increased, thus increasing the runtime of the tracking system. Once this change was made, however, the tracking performance did improve, as evidenced in Figures 7, 8, and 9. Alternatively, one could devise a cleverer proposal density that accounts for observed range ellipses when propagating particles in the time-update step, rather than simply use the prior; this is addressed in Section 7.2. When using a suboptimal filter – and a particle filter with a finite number of particles is necessarily suboptimal – exploiting more data may require more computation, or at least more sophisticated computation. Several approaches for designing a better proposal density are discussed in [16].

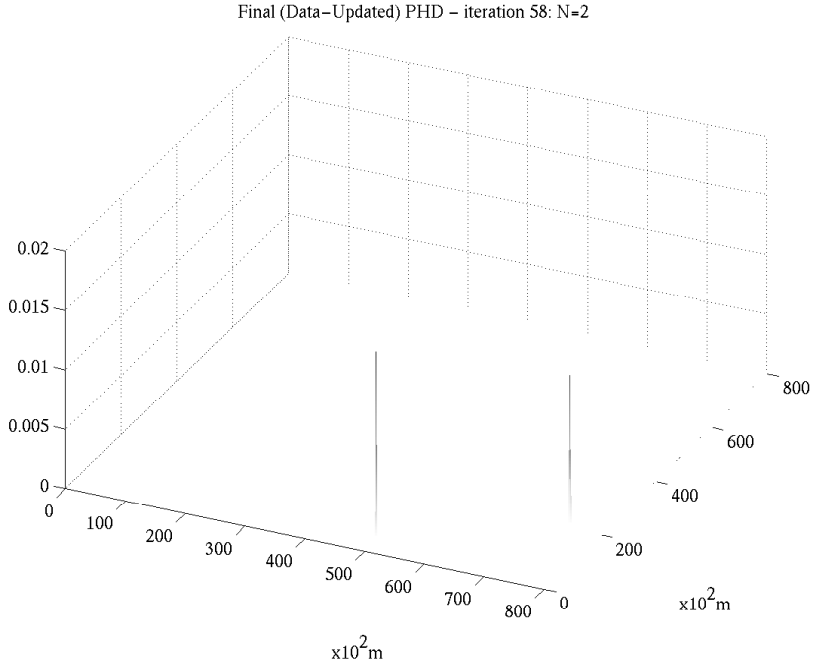
In the $p_{FA} = 10^{-2}$ case, including Doppler information allowed the filter to estimate the correct target number and track the targets, which it could not do with just the range-only measurements. However, since the simulation did not always consistently track both targets, a cleverer proposal density would also be expected to improve tracking performance in this case. All in all, the PHD filter was effectively able to incorporate the velocity observations and facilitate improved multitarget tracking performance and better target number estimation, but it appears that, in general, this will require either additional particles or smarter particles, or both. Each would require additional computation.

It merits a brief mention that the EM algorithm used is the largest consumer of simulation time in our multitarget tracker. With an alternative peak-extraction algorithm, one may be able to add the additional computation needed to achieve better performance but still retain reasonable computational efficiency. An alternative peak-extraction technique is presented in Section 5.2.4.

A nice avenue for future work would be to compare the PHD-based particle filter to other multitarget tracking methods in the passive radar context. The PHD is not so much a multitarget tracking technique as it is an easy multisensor fusion technique in the multitarget context. The PHD is appealing in that it would easily allow us to incorporate other types of measurements, such as angle measurements from an infrared search-and-track sensor or hyperbolic measurements from a time-difference-of-arrival electronic-support-measure (ESM) sensor, into the multitarget tracker. As was the case with velocity measurements, provided we have sufficient computing power, we would expect these additional measurements to improve the multitarget tracker's performance.

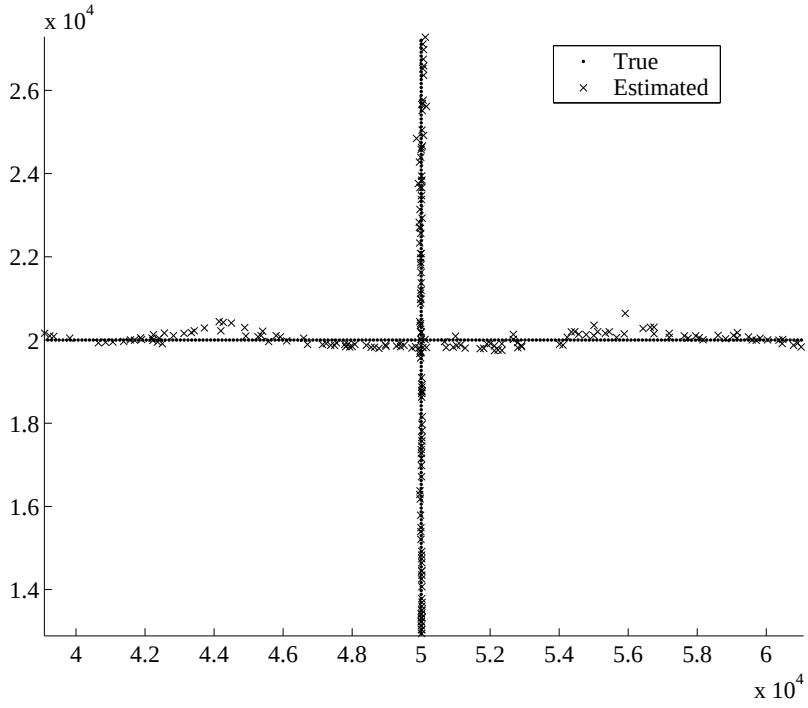



(a) The PHD particle filter and range ellipses. The receiving antenna is represented by the hexagon, and transmitting antennas by the triangles. The two diamonds indicate the actual target positions. Each particle of the filter is pictured. Note the targeted cluster placement of birth particles and the absence of persistent ghost targets.

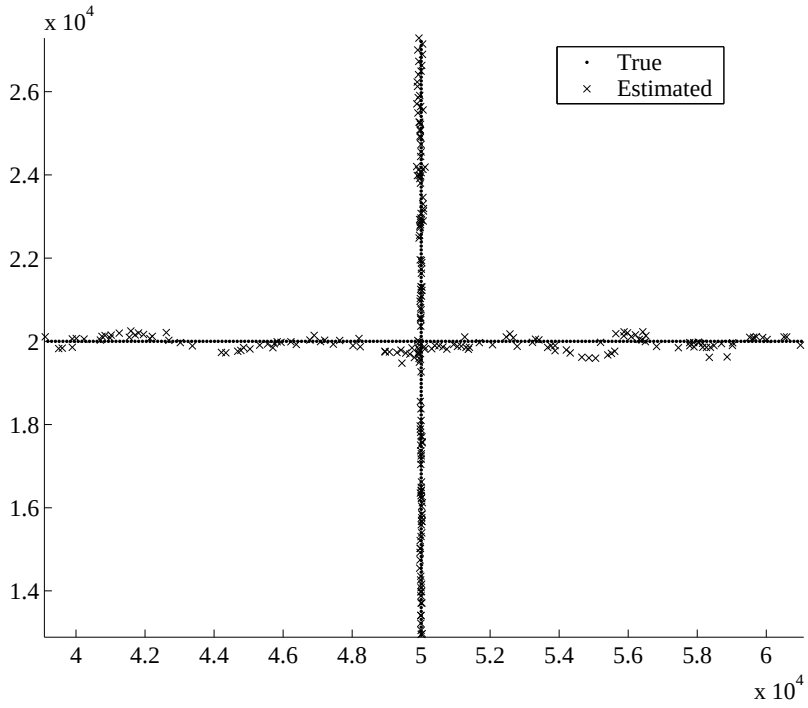



(b) The particle weights of the PHD filter. The sum of the weights is 1.997.

**Figure 5:** The 1,500 particle range and velocity PHD filter at time $k = 58$.




(a) Range and velocity particle filter




(b) Range-only particle filter

**Figure 6:** Close-up of range-only filter results and range and velocity filter results. Both are using 1,500 particles.

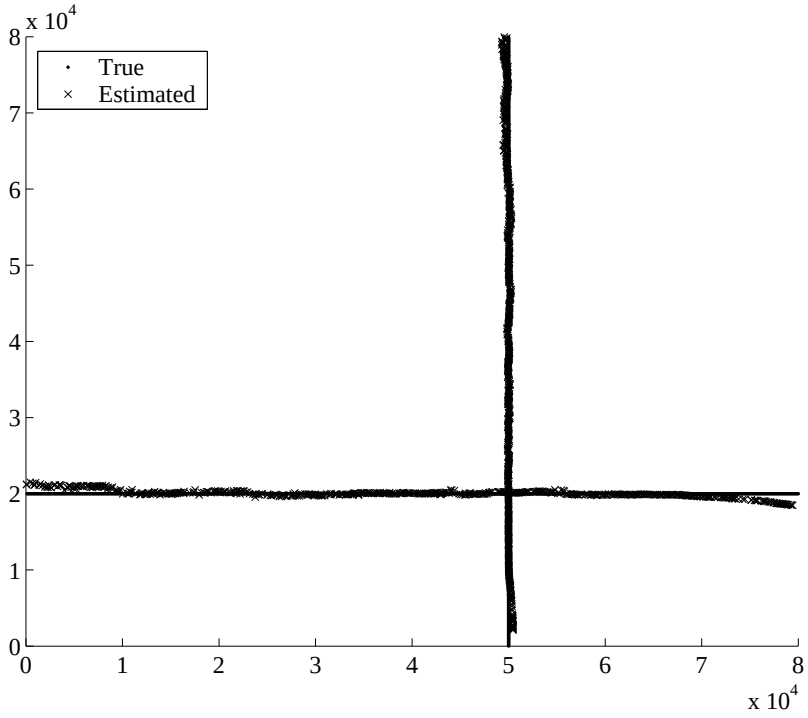



**Figure 7:** Actual vs. estimated target locations as given by the 3,000 particle range and velocity PHD filter.

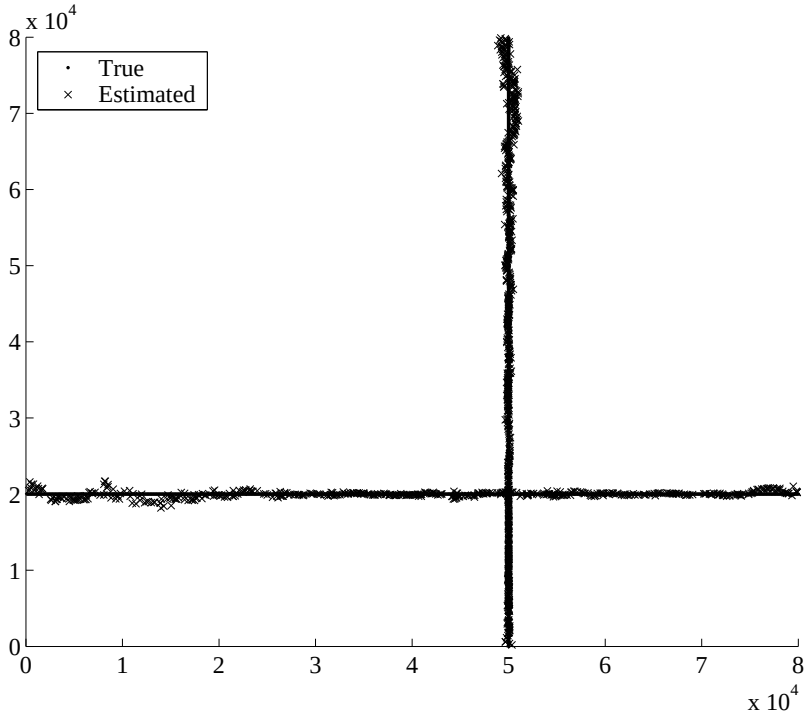
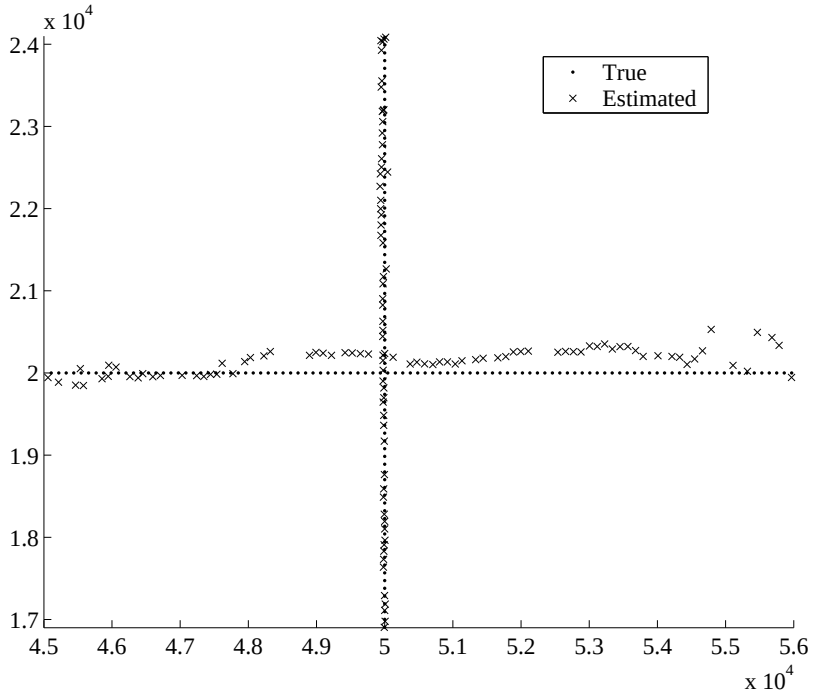
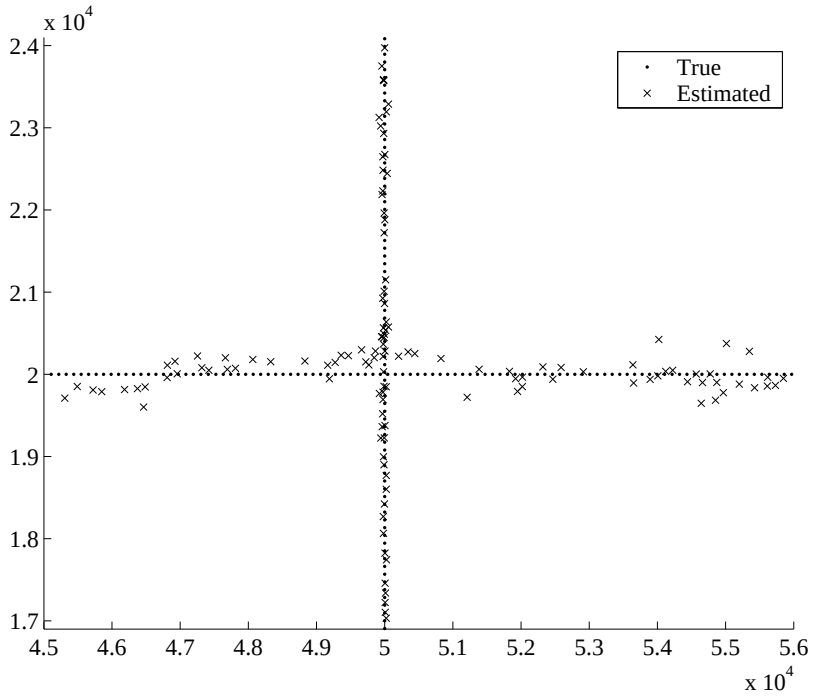



**Figure 8:** Actual vs. estimated target locations as given by the 3,000 particle range-only PHD filter.

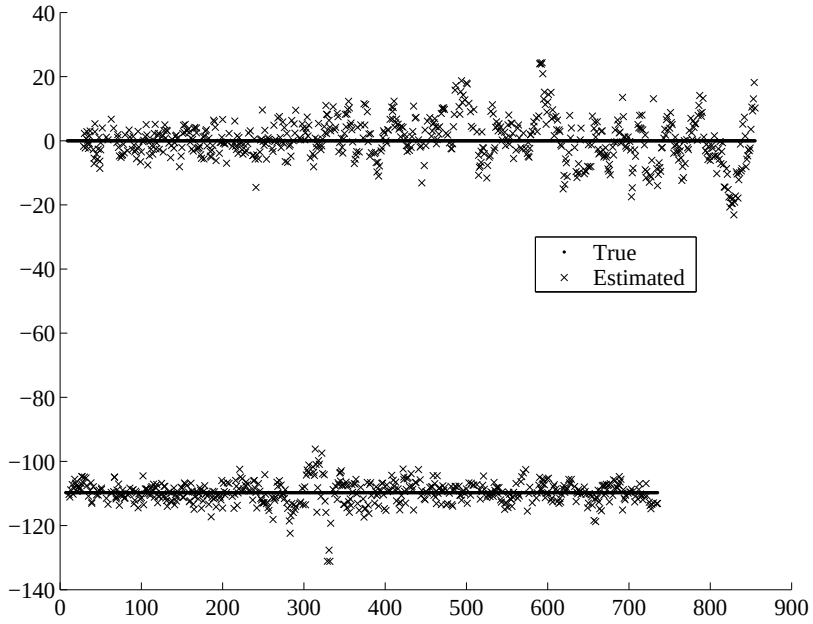



(a) Range and velocity particle filter

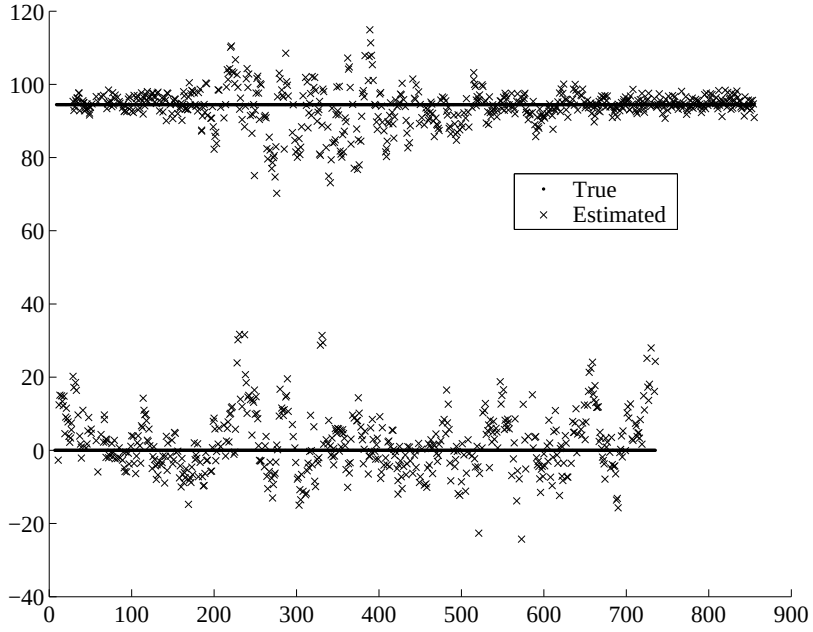



(b) Range-only particle filter

**Figure 9:** Close-up of range-only filter results and range and velocity filter results. Both are using 3,000 particles.

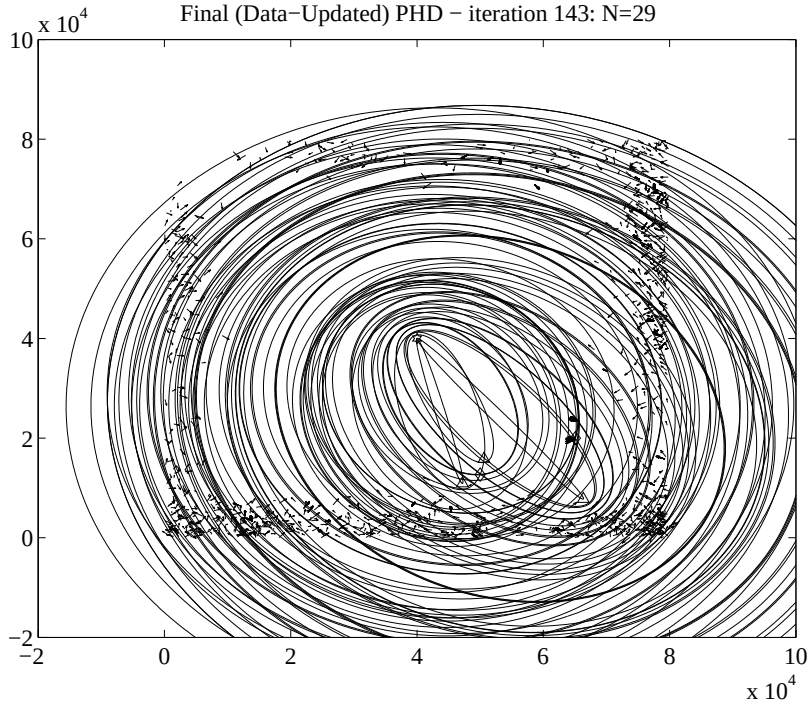



(a) Target $\dot{x}$ components

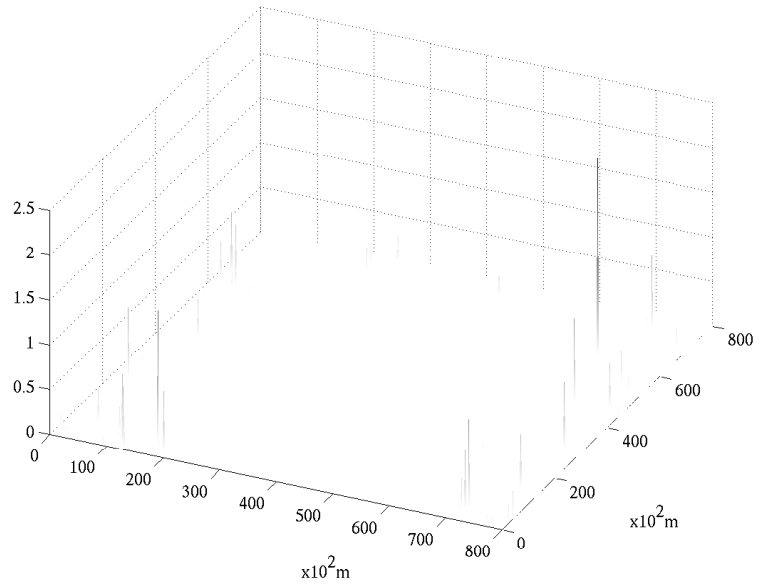



(b) Target $\dot{y}$ components

**Figure 10:** Actual vs. estimated target velocities as given by the 3,000 particle range and velocity PHD filter.




(a) The PHD particle filter and range ellipses. The receiving antenna is represented by the hexagon, and transmitting antennas by the triangles. The two diamonds indicate the actual target positions. Each particle of the filter is pictured.




(b) The particle weights of the PHD filter. The sum of the weights is 29.47.

**Figure 11:** The 3,000 particle range-only PHD filter at time $k = 143$ when $p_{FA} = 10^{-2}$. There are 81 false alarm observations. The filter estimates 29 targets present.

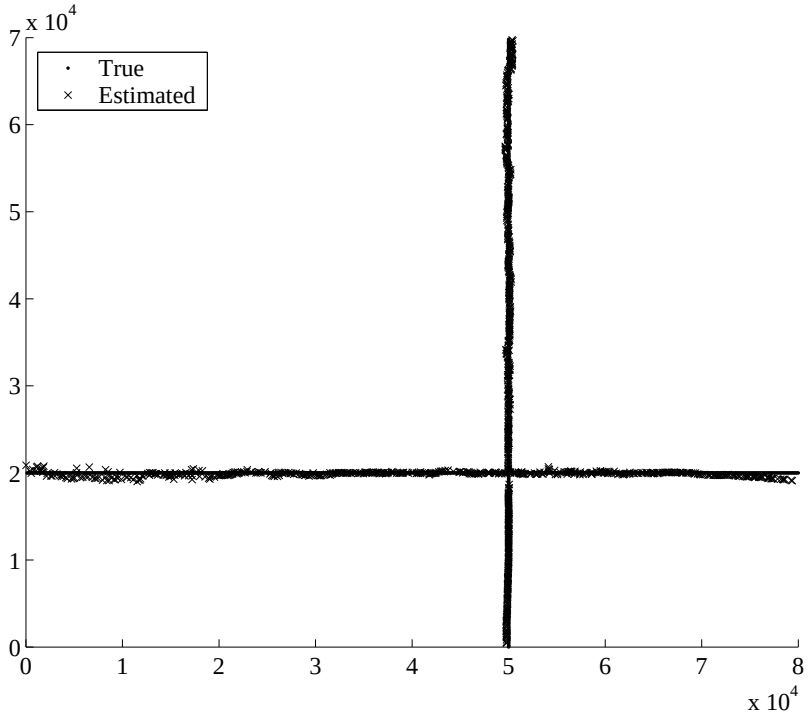
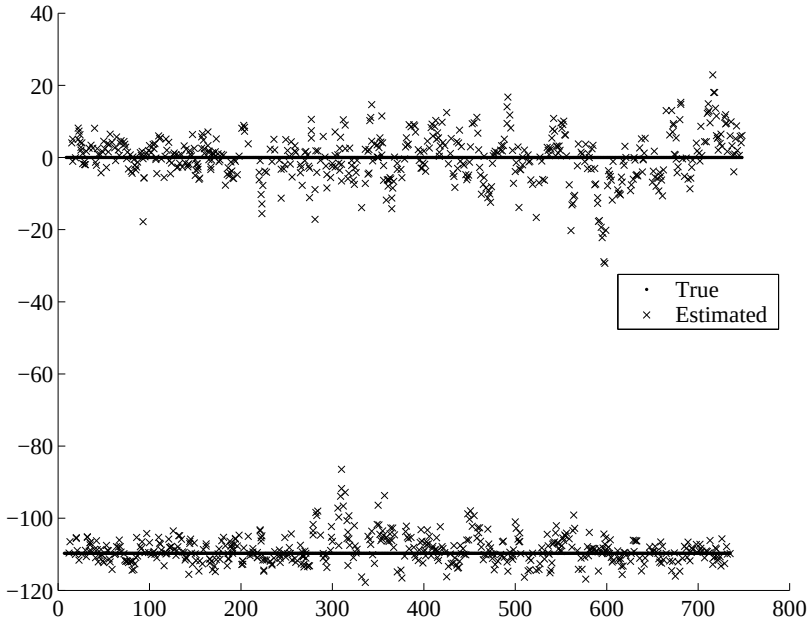
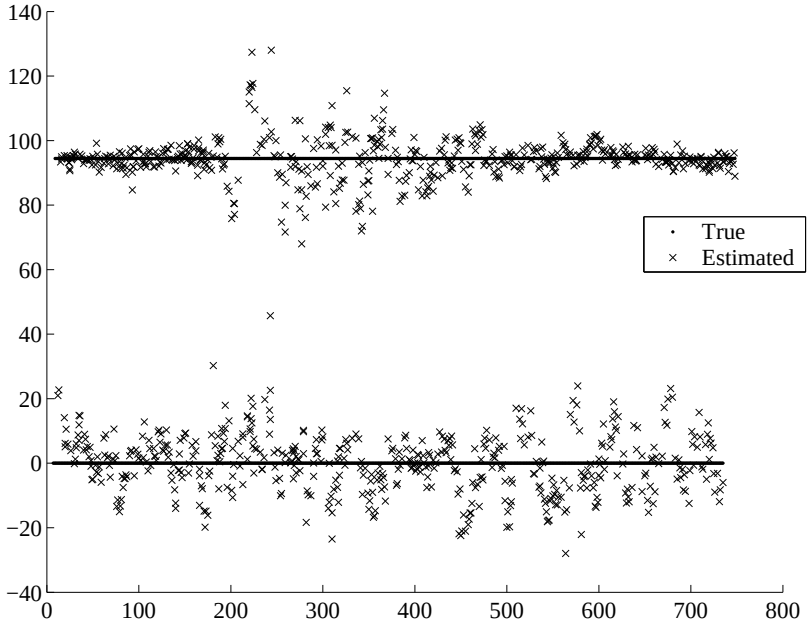



**Figure 12:** Actual vs. estimated target locations as given by the 3,000 particle range and velocity PHD filter when $p_{FA} = 10^{-2}$.




(a) Target $\dot{x}$ components




(b) Target $\dot{y}$ components

**Figure 13:** Actual vs. estimated target velocities as given by the 3,000 particle range and velocity PHD filter when $p_{FA} = 10^{-2}$.

# CHAPTER V

# MULTITARGET, MULTISENSOR TRACKER: SUBSEQUENT IMPLEMENTATION

## *5.1 Introduction*

Thanks to Dr. Paul Howland, I was fortunate to have the opportunity to intern at the NATO Consultation, Command and Control Agency (NC3A) in The Hague, Netherlands, during the summer of 2005. While there, I was able to improve the functionality of the particle-filter implementation of the PHD Bayes filter presented in Chapter 4. The implementation presented in this chapter is a realistic multitarget tracking simulation using the passive radar configuration at NC3A. Unfortunately, real data was unobtainable due to issues with the receiver hardware.

## *5.2 Improvements to the Multitarget, Multisensor Tracker*

The following section contains a list and description of the work accomplished at NC3A to improve the multitarget, multisensor PHD-based tracker for passive radar.

### 5.2.1 Number of Particles

The number of particles in the system were modified to be dependent on the estimated number of targets present. In the Washington, D.C. simulation in Chapter 4, the number of birth particles and resampled particles were each specified as fixed parameters. This was changed to the more common and computationally-efficient method used in particle filters, whereby the number of targets present determines the number of particles to be used. In the case of too few targets, a minimum number of particles can also be specified. The number of birth particles at each time step is still roughly fixed at each iteration. The number of particles used in the NC3A scenario is given in Table 13.

### 5.2.2 Sensor Data Collection

Hence, the simulation was modified so that only one sensor provides observations at each time step. The simulation waits for all sensor reports to be collected before running the PHD filter. This was done to match how the real passive coherent location (PCL) system currently works. That is, the receiver collects data over a processing interval on the frequency of one of the transmitters, after which it re-tunes to the frequency of the second transmitter to collect data, and then to the third transmitter, and back to the first, and so on. Note that this requires the observations from the first two transmitters to be propagated in time according to the target motion model, so that they are still sufficiently valid when the PHD filter is run.

### 5.2.3 Antenna Pattern

The real antenna gain pattern was incorporated into the simulation to replace the simple omnidirectional antenna model for the receiver. Though the antenna gain pattern varies slightly depending on the frequency of the received signal, the gain pattern used in the simulation is for the case where the receiver is tuned to 98 MHz. The antenna gain pattern is shown in Figure 14. Incorporating the real antenna gain pattern affects the signal-to-noise ratio (SNR), via the $G_R$ term in (123), and those parameters that depend on SNR, such as range variance and probability of detection. A bird's-eye view of the NC3A transmitter geometry and of the antenna pattern for the area in which all transmitter-receiver pairs have high SNR can be seen in Figure 19(b). The result for each transmitter-receiver pair is shown in Figures 17 and 18. Note the big nulls introduced at $-90°$ and $90°$ off boresight. Unfortunately, the orientation of the antenna pattern is not specified in the data plots used, so it is possible that the real antenna pattern is in fact a mirror image across boresight of the pattern implemented in the simulation.

### 5.2.4 Peak Extraction

New peak extraction logic was created to be used in lieu of the EM algorithm. As noted in Sections 4.5.2 and 4.6.2, the EM algorithm used to extract the peaks from the PHD

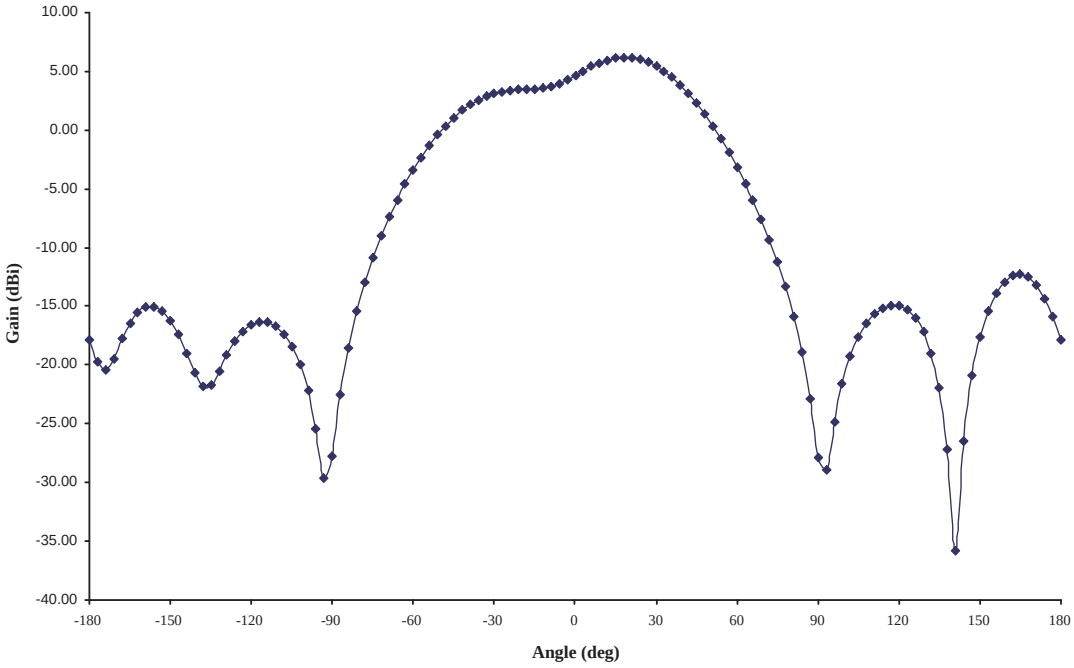



**Figure 14:** The antenna gain pattern of the NC3A receiver. The gain pattern is for the E plane of the dipole array, where one dipole is fed and the other terminated. The frequency used is 98 MHz. Plot courtesy of Paul Howland.

exhibited undesirable behavior. Apart from having a considerably long execution time, the EM algorithm frequently failed to find the correct peak locations. The algorithm often aborted because it produced a singular covariance matrix or because it took too many iterations to fit the Gaussians to the PHD. Other times, it would "double-fit" Gaussians to a single peak, thereby extracting a single target twice. Essentially, the poor performance of the algorithm was due to our attempt at fitting a Gaussian mixture to a density that was not a Gaussian mixture.

A new peak extraction algorithm was developed that is similar to the CLEAN technique used in astronomy to uncover secondary objects in an image by removing the effects of the primary objects [9]. The new peak extraction technique takes advantage of the properties of the PHD, and it works more accurately and much faster than the EM algorithm. It takes the highest peak in the PHD as the target location, and then extracts a target's worth of weight from the PHD at and around this point before searching for the next highest peak. Thus, the property of the PHD that it integrates to the expected number of targets

is exploited. Pseudo-code for this new peak extraction algorithm is provided in Table 10.

Unlike the EM algorithm, the new algorithm is guaranteed to return a state value for every peak it is told to extract. This is because it simply returns the state of the particle with the largest weight when looking for a peak. It then constructs a neighborhood, based on the range and Doppler resolutions, around that particle in the state space. The region of the neighborhood is increased until the sum of the weights of the particles inside it are equivalent to a target's worth of weight. The weight is then subtracted from the region – thereby, effectively extracting the target – and the procedure is repeated for the next desired peak. Occasionally, especially when the algorithm is extracting the final desired peak in a time step, the actual weight in the neighborhood around the peak is not sufficient to sum up to a target's worth. In this case, the algorithm still simply reports the target state of the peak and continues as normal.

The new peak-extraction algorithm was found to perform better, and significantly faster, than the EM algorithm. The new peak-extraction technique was also compared to a $k$-means clustering algorithm, and it provided results that were as good, if not better, than the $k$-means algorithm. The new peak-extraction algorithm performed better in the case where there were two peaks in the PHD but where only one target was estimated as being present. In this case, the $k$-means algorithm produced a target detection that was located somewhere between the two peaks; whereas, our peak extraction algorithm extracted the single target correctly.

**Table 10:** Pseudo-Code for Peak Extraction Algorithm

*Compute radius vector, $\boldsymbol{\rho}$, to use for determining region of peak extraction:*

- $\boldsymbol{\rho} = [\Delta_R, \Delta_R, 2\Delta_D, 2\Delta_D]^{\mathsf{T}}$, where
- $\Delta_R = \frac{c}{\beta_{min}}$ is the largest range resolution among the transmitters, and
- $\Delta_D = \frac{\lambda_{max}}{CPI}$ is the largest Doppler resolution (in m/sec) among the transmitters.

*Determine number of peaks to extract and calculate weight of each peak. The rounding error is assumed to be distributed evenly over all targets, and a 1% margin is added to the weight extraction:*

- Expected number of peaks = round $(\sum w_{i,k+1})$.
- Target weight = $\min\left(0.99,\ 0.99 \cdot \frac{\sum w_{i,k+1}}{\text{Expected number of peaks}}\right)$.
- Let $w_i$ be a copy of $w_{i,k+1}$, which are the particle weights in the PHD filter at time $k+1$.

  Note that the set of $w_i$ weights are modified in the course of the peak extraction. The actual $w_{i,k+1}$ in the PHD filter are unaltered by the peak extraction algorithm.

*Extract peaks:* for $p = 1$ to Expected number of peaks,

- Extracted peak = $\boldsymbol{\xi}_{j,k+1}$, where $j = \arg_i \max(w_{i,k+1})$.
- Let $w_{max} = w_j$.
- Number of peaks found = Number of peaks found + 1.

- If $w_j \geq$ Target weight, then set $w_j = w_j -$ Target weight.

  Return $\boldsymbol{\xi}_{j,k+1}$ as the extracted peak. Continue with the $p$-for-loop to find the next peak.
- Else, for $n = 1$ to MaxTries,
  - Neighborhood = $\boldsymbol{\rho} \cdot n$.
  - Find all particles $\boldsymbol{\xi}_{\alpha,k+1}$ and their corresponding weights $w_\alpha$, that are in the given neighborhood of $\boldsymbol{\xi}_{j,k+1}$. That is, find $(\boldsymbol{\xi}_{\alpha,k+1}, w_\alpha)$, such that:

    $$\boldsymbol{\xi}_{\alpha,k+1} \in [\boldsymbol{\xi}_{j,k+1} - n\boldsymbol{\rho},\ \boldsymbol{\xi}_{j,k+1} + n\boldsymbol{\rho}].$$

  - Neighborhood weight = $\sum_\alpha w_\alpha$. (Note that $w_{max}$ is included in the Neighborhood weight.)
  - If Neighborhood weight $\geq$ Target weight, or $n =$ MaxTries, then a peak has been found or the current peak extraction is being cut short. In either case, reduce the weight of the particles in the Neighborhood by the Target weight:
    - Set $w_j = 0$, since $w_{max} \leq$ Target weight.
    - Set

      $$w_\alpha = w_\alpha \cdot \left(1 - \frac{\text{Target weight} - w_{max}}{\text{Neighborhood weight}}\right),$$

      such that the weight in the neighborhood is reduced by a target's amount of weight, not including the weight already removed at $w_j$. Note that the weight is removed proportionally, so that the peak structure in the particles is preserved.
    - Return $\boldsymbol{\xi}_{j,k+1}$ as the extracted peak. Continue with the $p$-for-loop to find the next peak.
  - Else, a target's amount of weight does not exist in the current neighborhood. Continue with the $n$-for-loop to expand the neighborhood.

### 5.2.5 Track Association: Nearest Neighbor

A simple nearest-neighbor association was added to associate the extracted PHD peaks with tracks. The implementation tries to assign a peak-extraction at each time step to a track of peak extractions performed during previous time steps. The nearest observation to a track is associated with it if the observation falls within a range-resolution distance of the track. Any unassociated peaks are assigned immediately to a new track. No "potential-track" logic was implemented. A track is dropped if it has not matched with any peaks within the timespan of three PHD-filter iterations.

### 5.2.6 Track Association: Track ID

As an alternative to the nearest-neighbor association post-processing algorithm, a "track ID" parameter was added to the particles, so as to provide an inherent way of identifying targets. This, however, did not prove successful.

In the PHD particle filter, the PHD is represented by the particle weights. However, when we propagate the particles according to the target motion model in the Time Update step, and when we locate the targets via the peak extraction of the posterior PHD (see Sections 3.2.3, 4.3.5, and 5.2.4), we treat the particle states, in effect, as target tracks. It was thought that if each particle were tagged with a track ID, then no additional track association logic after extracting the peaks would be necessary. The PHD particle-filter, as it stands, does not maintain any association between a peak extracted at time $k$ and a peak extracted at time $k+1$. Thus, this was an attempt to insert explicit track logic into the PHD.

The implementation was as follows. At each time step, the birth particles were all assigned a track ID unique from any existing particle's track ID. The existing particles simply kept their ID as they were propagated in the time-update step. After peak extraction, the track ID of the largest weighted particle in the peak was considered to be the ID of the extracted target. Another method of assigning the ID was also attempted, whereby the statistical mode of the target IDs of all the particles in the extracted peak was considered to be the ID of the extracted target. However, this gave poorer results than simply extracting

the peak target ID.

Unfortunately, even using the better of these two methods, the implementation produced poor results. An additional improvement to the implementation that was thought to be potentially able to render this tracking logic successful is discussed in Section 6.2; alas, we found that this did not work well, either.

### 5.2.7 Regions of low $p_D$

Logic was added to remove particles completely (both propagated and birth) if they fall in regions where any of the sensors have a $p_D < 0.95$, or if they fall outside the field of view (FoV), since we only want to work in areas of high SNR. As discussed in Section 4.6.1, particles found in areas of low SNR are also in regions of low $p_D$ and cause the filter to overestimate continually the number of targets present. This is because the filter has no reason to doubt that birth particles in these areas belong to valid targets, regardless of actual observations, since the filter simply assumes that it just cannot see the targets in these areas due to the low $p_D$. We address this issue again in Section 6.3.

## *5.3 Improved Birth Particle Placement*

Section 4.3.3 described how the birth particles in the Washington, D.C. scenario were distributed either uniformly around the edges of the FoV or in a clustered placement at the points where the bistatic ellipses intersected the edges of the FoV. However, as mentioned in the results and conclusions of that implementation (see Sections 4.5 and 4.6.2), the birth particle placement needed to be replaced with a smarter proposal function, since it would often fail to detect the targets. A smarter proposal for the birth particles is described below, in which birth particles are placed at the intersections of the bistatic range ellipse observations. As part of this improvement, the logic for placing birth particles around the edge of the FoV, either uniformly when no ellipse intersections are present or in a clustered fashion when ellipses are present, was removed. This change was also needed to avoid the target overestimation problem caused by particles being present in areas of low SNR.

We tried two techniques for implementing the smarter birth particle placement. The first focused on range resolution, and the second focused on range variance. We found that

the second technique worked better.

### 5.3.1 The Range-Resolution Based Grid Technique

In an attempt to more cleverly distribute birth particles, ellipse-intersection-counting grid logic, based on range resolution, and recursive, combinatorial, Doppler-intersection finding logic were added to our multitarget tracker. They localize the placement of the birth particles to the intersections of the bistatic range ellipses and initialize the birth particles with appropriate velocity parameters.

The counting-grid on which to localize the ellipse intersections is constructed by forming overlapping gridspaces, each of which has a width equivalent to the largest bistatic range resolution among the three transmitters. Adjacent gridspaces overlap by 50%; that is, the edges of a gridspace's neighboring gridspaces pass through its center (see Figure 15). During an iteration of the PHD filter's time-update step, the number of bistatic range ellipse observations from different transmitters that pass through each gridspace is tallied. Only those gridspaces that have a count of three or more are considered candidates for receiving birth particles. This logic, thus, takes care of localizing the ellipse intersections in range. In other words, the ellipse intersections have been narrowed down to local regions in the $(x, y)$ subset of the target state space.

Now, using only those ellipses that intersect in range, a least-squares solution is implemented to find the possible velocities associated with their Dopplers, and the intersections are then narrowed down further using this information (see Section 5.3.4). The result is a set of gridspaces that contain bistatic ellipse observations that intersect in both range and Doppler and, thus, contain possible targets. The birth particles are then placed uniformly with respect to their $x$ and $y$ positions in these gridspaces. To initialize their velocity parameters, the particles are distributed uniformly in a region of

$$[\hat{v} - 2\Delta_{\dot{R}}, \hat{v} + 2\Delta_{\dot{R}}], \tag{142}$$

where $\Delta_{\dot{R}}$ is the Doppler resolution (see (134)), and $\hat{v}$ is the least-squares solution for the velocity observation in the gridspace. The birth particles are divided evenly among the valid target-observation gridspaces. Thus, since the total number of birth particles introduced

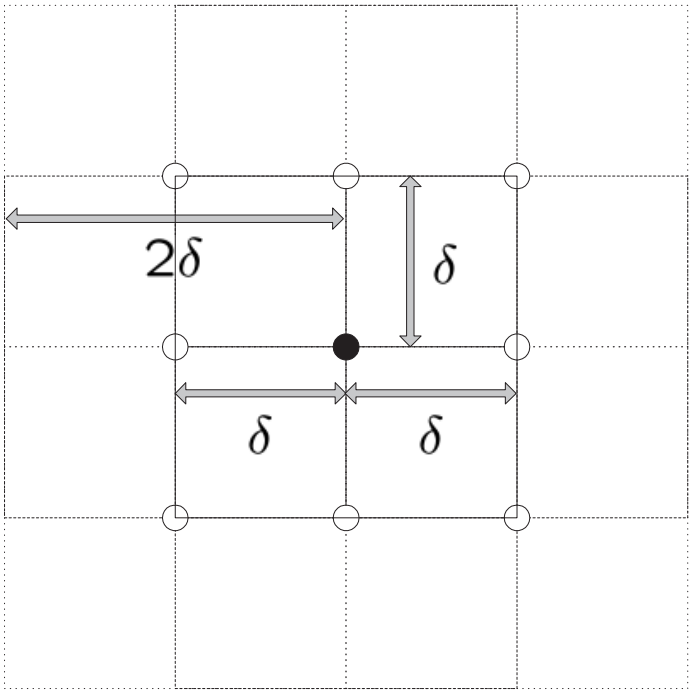



**Figure 15:** The gridspacing used in the offline grid. The figure depicts a central gridspace of size $2\delta \times 2\delta$ surrounded by eight adjacent gridspaces. The centerpoint of each gridspace is indicated by a circle. The center gridspace is filled in. In the range-resolution based grid technique of Section 5.3, $\delta = \frac{\Delta_{R,max}}{2}$, where $\Delta_{R,max}$ is the largest range resolution among the three transmitters. In the range-variance based grid technique, $\delta \approx 2\sigma_R$, as described in Section 5.3.3.

into the filter is fixed as a configurable parameter, the number of birth particles placed in each "potential-target" gridspace depends on the number of such gridspaces, since each receives an equal share of the total number of birth particles allowed. (Future work could explore allowing the total number of birth particles to vary with the number of "potential-target" gridspaces.) The total weight of all the birth particles sums to the number of birth targets expected at the particular time step. In our simulations, this is assumed to be one.

Note that the bistatic range resolution remains constant throughout the simulation, so the counting grid can be computed offline before the simulation begins. This makes this grid method practical for near real-time simulation, since computing such a grid at each time step would be impractical.

### 5.3.2 Result of the Range-Resolution Based Grid Technique

Even though the range resolution-based grid method appeared to work reasonably well, it was not robust relative to changes in SNR. This is because the likelihood model that determines the weights in the PHD filter (see Section 4.4.3) is dependent not on the range resolution, but on the range variance, which is itself dependent on SNR. Thus, for example, if the SNR were to increase, the range variance then tightens, and the range-resolution based grid no longer positions the particles close enough to the regions of high importance. This was evident when the SNR was increased by a factor of 100. The range variance accordingly shrank by a factor of 10, and the simulation was no longer able to track a target that it had previously tracked. However, when the number of birth particles was also increased by a factor of 100 (needed since we require a factor of 10 increase in both the $x$ and the $y$ dimensions in the target state - hence $10 \times 10 = 100$), the filter was able to track the target again. This demonstrates that the range-resolution grid method does not solve the birth particle placement problem, since target detection, and not just tracking accuracy, is still dependent on having a large number of particles.

### 5.3.3 The Range-Variance Based Grid Technique

To solve the problem described in the previous subsection, the grid needs to have gridpoints (i.e., grid centerpoints) that are spaced according to range variance. The difficulty with this

is that range variance varies with position in the FoV. Thus, the grid will not have evenly spaced gridpoints, as was the case when using range-resolution, but rather it will have gridpoints that are more closely spaced in regions of high SNR (i.e., tight $\sigma_R$). Fortunately, this variable grid can be computed offline, since the receiving and transmitting antennas are immobile. The grid was formed by placing most gridpoints at a distance of $2\sigma_R$, and others no closer than $1.7\sigma_R$, from each other (see Figure 15). This flexibility in separation distance was needed to obtain full coverage of the FoV with variably-sized grid spaces, since the algorithm that creates the grid would otherwise mop itself into a corner before it had placed gridspaces throughout the whole FoV. Also, to prevent the grid-making algorithm from being sucked into a black hole around the antennas (where SNR is increasingly large and $\sigma_R$ increasingly small), it stopped placing gridpoints when their gridspace width was less than 10 meters. This limit on gridpoint placement, however, also contributed to the "mopping into a corner" issue of the grid creation algorithm. The resulting grid is shown in Figure 20.

Each gridspace consists of a $4\sigma_R \times 4\sigma_R$ region centered around a gridpoint. Thus, the gridspaces overlap as they did for the range-resolution based grid to obtain adequate coverage. Note that the value of the SNR at the gridpoint is used to determine the $2\sigma_R$ distance to the next gridpoint, and the smallest $\sigma_R$ among the three possible $\sigma_R$ values from the three transmitters is used, since the PHD weights are ultimately affected by all three transmitter's $\sigma_R$ values at any given location. Note the chicken and egg problem evident in our using the SNR at the gridspace's center instead of the largest possible SNR in the gridspace. We can't use the largest possible SNR, because we don't yet know what size the gridspace will yet be, since it is calculated from the SNR!

Also evident in the grid construction is a tradeoff between whether to use the smallest of the three possible $\sigma_R$ values to determine the width of each gridspace, or to use the largest of the three. It would be desirable to use the largest of the three values and have large gridspaces, so that no ellipse intersections are missed due to noisy range observations. However, it is also desirable to use the smallest of the three $\sigma_R$ values, so that targets can be localized adequately and particles distributed in a small area of highest importance, since

even if an ellipse intersection is missed due to the observations being too far apart, the intersection would have been given low weight by the PHD filter had it been detected, in any case. Having better target localization is important when there are many false alarms present.

There is also another consideration involved when deciding whether to use the larger gridspace or the smaller one. There will be fewer ellipse observations passing through the smaller gridspace, and so the runtime of the combinatorial Doppler-intersection logic will be shorter. However, because the gridspace is smaller, there will be more gridspaces to process, and so more runs of the Doppler-intersection logic will be required. A detailed analysis of the computational tradeoffs has not been performed, but rather we focus on the matter of target localization, and so we proceed to use the smallest $\sigma_R$ value among the transmitters at each gridpoint for the gridspace.

While constructing the grid offline, the pseudo-inverse matrices needed by the Doppler-intersection logic to compute the least-squares Doppler solutions at each gridpoint are also computed. The Doppler-intersection logic is described in Section 5.3.4.

### 5.3.4 Doppler-Intersection Logic: The Least-Squares Solution

The grid method works by identifying all the grid spaces through which more than three ellipses (from different transmitters) pass. This finds the $x$ and $y$ components of the ellipse intersection. To find the $\dot{x}$ and $\dot{y}$ components, every combination of possible triplets of ellipses in the gridspace is formed, and each triplet's associated Doppler observations are used to compute its least-squares solution for target velocity. These velocity estimates are then used to obtain Doppler observation estimates, and the error residual between these estimated Dopplers and the actual observed Dopplers is computed. Only those ellipse triplets whose Doppler error residuals are within a given threshold, which is dependent on the Doppler resolutions of the transmitters, are retained as valid ellipse intersections. The least-squares solution algorithm is as follows.

The rate of change in bistatic range that is observed by the radar can be computed (see

(110)) from a target's position and velocity:

$$\dot{R} = \frac{(x-x_r)\dot{x}+(y-y_r)\dot{y}}{R_r} + \frac{(x-x_t)\dot{x}+(y-y_t)\dot{y}}{R_t} \tag{143}$$

$$= \left(\frac{(x-x_r)}{R_r} + \frac{(x-x_t)}{R_t}\right)\dot{x} + \left(\frac{(y-y_r)}{R_r} + \frac{(y-y_t)}{R_t}\right)\dot{y} \tag{144}$$

$$= \quad C_t\,\dot{x} \quad + \quad D_t\,\dot{y}, \tag{145}$$

where $R_r$ is the distance between the target and the receiver, and $R_t$ is the distance between the target and transmitter $t$. Note that $C_t$ and $D_t$ can be computed offline when using a grid system, since $x$ and $y$ are simply the location of the centerpoint of the gridspace.

Thus, with a triplet of PCL-generated Doppler observations, we have an overconstrained system:

$$\begin{pmatrix} C_1 & D_1 \\ C_2 & D_2 \\ C_3 & D_3 \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} \dot{R}_1 \\ \dot{R}_2 \\ \dot{R}_3 \end{pmatrix}$$

$$A \qquad v \quad = \quad b, \tag{146}$$

where $\dot{R}_t$ is the Doppler observation from the $t$-th transmitter, $C_t$ and $D_t$ are computed offline as given above, and $\dot{x}$ and $\dot{y}$ are the target velocity components that we are trying to find. Using the notation in (146), the least squares velocity estimate is

$$\hat{v} = A^T b = (A^T A)^{-1} A^T b. \tag{147}$$

Note that $(A^T A)^{-1} A^T$ is precomputable offline. We now plug $\hat{v}$ back into (146) to obtain the $\dot{R}$ estimate we would receive if the target located in the gridspace actually were to have this velocity estimate as its true velocity:

$$\hat{b} = A\hat{v}. \tag{148}$$

The error vector residual is computed as:

$$e = \hat{b} - b. \tag{149}$$

An ellipse triplet is then marked as a valid intersection if the sum of the squares of the components of its error vector, $e$, is less than the sum of the squares of twice the Doppler resolution for each transmitter at the gridspace center.

Note that the Doppler resolution (134) is used here, rather than the Doppler variance (127). This is because the smallest value of the Doppler variance is limited by the value of the Doppler resolution. That is, the Doppler variance (or, more correctly, standard deviation) can be no less than the Doppler resolution times the wavelength used. Thus, unlike the range variance, which can become much smaller than the range resolution and requires the more complicated variably-spaced grid method to be used, the Doppler observation does not give us the same difficulty. Thus, the simpler Doppler resolution is used.

### 5.3.5 Alternative Intersection-Finding Logic: The Iterative Least-Squares Approach

Another approach to the birth particle placement problem is to use a least-squares iterative technique which solves for both range and Doppler intersections. This method has already been implemented by Kees Stolk, a colleague at NC3A. It does not require an offline computation of a grid, but does require an exhaustive combinatorial search over all the observations at a time step for the ellipse intersections.

Compared to the grid method, the iterative method might require more computation when there are few gridspaces containing three or more ellipse intersections. However, because the grid spaces overlap by half their width, it is possible to have the same ellipses intersect in more than one gridspace. This causes redundant computational work in the grid approach, since the simulation currently searches for all possible ellipse combinations in each gridspace to find the Doppler solutions. Thus, the gridspace technique may at times require more computation than even the exhaustive iterative least-squares approach. A better manner of overlapping the gridspaces, instead of using a simple $2\sigma_R$ spacing, may improve its runtime efficiency.

A problem with the iterative least-squares approach is that it does not always converge to the correct intersection. As illustrated in Figure 16, the algorithm may converge to a ghost target depending on the initial starting point of the algorithm. Additional information, such

as angle of arrival observations, would be needed to correctly initialize the algorithm. This problem is not an issue when using the grid approach to find the ellipse intersections, since all valid intersections will be treated as regions of potential targets in the grid method, and the PHD filter will handle the task of distinguishing the ghost targets from the real ones.

## *5.4 Results*

This section presents the results of the NC3A simulation, given the improvements and modifications presented in Section 5.2 and using the variably-sized grid method introduced in Section 5.3.3. Tables 11 and 12 provide the details on the transmitters and receiver used. The receiver is assumed to be located at NC3A in The Hague, Netherlands. The "$x$-dist." and "$y$-dist." are the distances of the transmitters from the receiving antenna in $x$ and $y$, respectively. The receiver parameters are assumed to be the same as for the Washington, D.C. scenario in Section 4.3. Note, however, that the $G_R$ parameter has now been replaced by the real antenna gain pattern. Additional simulation parameters are given in Table 13. Note the considerable reduction in the number of particles needed from that of the Washington, D.C. simulation. The specified probability of false alarm generates about one to six false alarms per time step.

**Table 11:** Transmitting Antenna Specifications in the NC3A Simulation

| Location | $x$-dist. | $y$-dist. | Frequency ($f$) | Power ($P_T$) | Bandwidth ($\beta$) |
|---|---|---|---|---|---|
| Lopik | 49,844 m | -10,127 m | 96.8 MHz | 100.0 kW | 45 kHz |
| Wieringermeer | 117,212 m | 91,179 m | 97.1 MHz | 50.0 kW | 45 kHz |
| Goes | -30,590 m | -65,922 m | 99.8 MHz | 50.0 kW | 45 kHz |

**Table 12:** Receiver System Specifications in the NC3A Simulation

| | |
|---|---|
| Coherent Processing Interval ($CPI$) | 1.0 sec |
| Reference Temperature ($T_0$) | 290 K |
| Noise Figure ($NF$) | 30 dB |

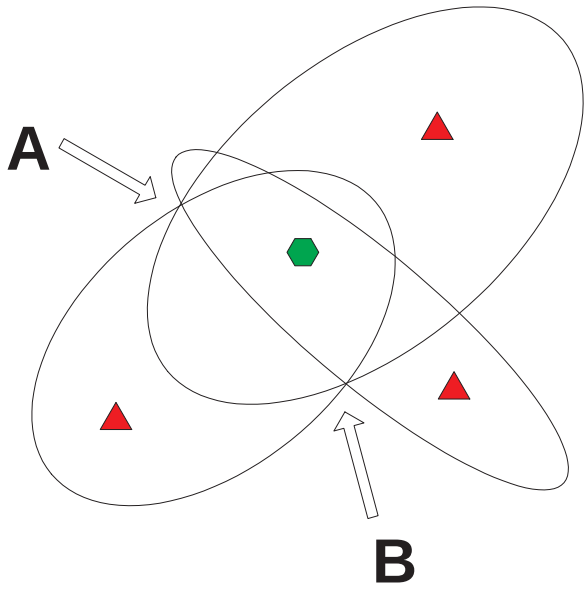


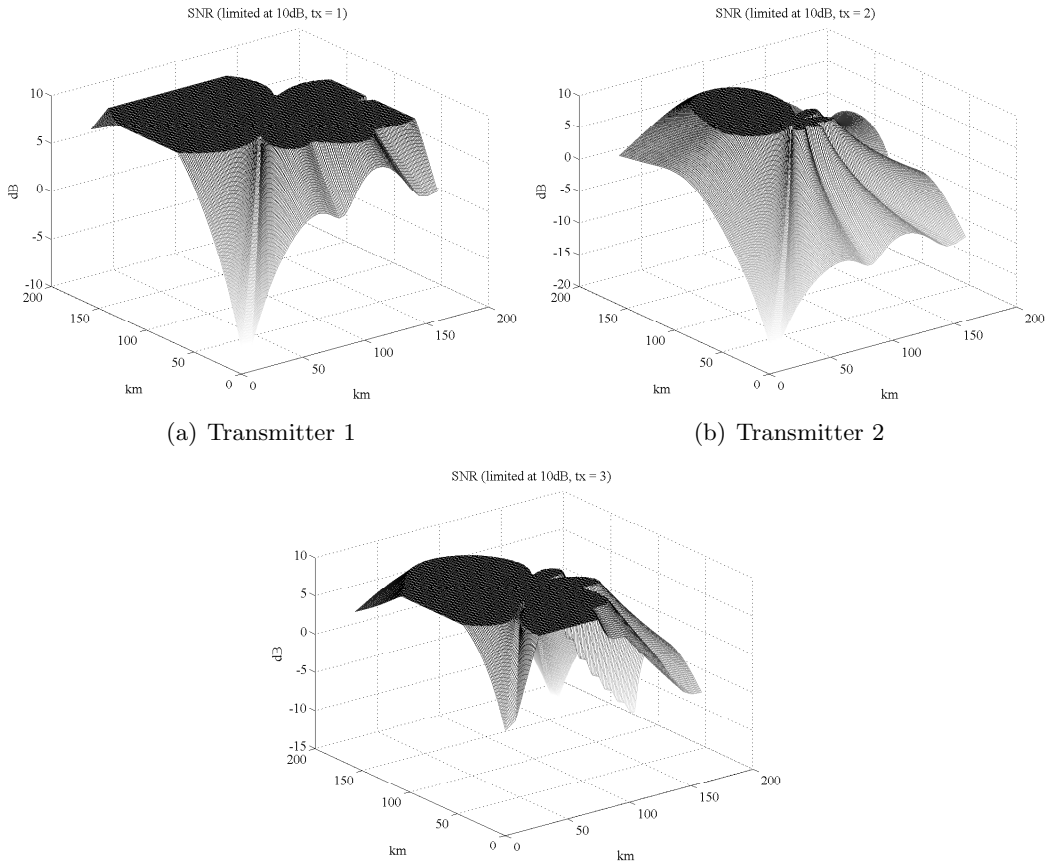

**Figure 16:** The iterative least-squares algorithm may converge to either point A or point B, depending on the algorithm's initial starting point. Three bistatic range ellipses are present. The receiver is represented by the hexagon, and the transmitters by the triangles. The configuration shown was created for illustration purposes and is not an exact representation of the simulation scenario.

**Table 13:** NC3A Simulation Parameters

| | |
|---|---|
| Simulated time period | 1 – 300 sec. |
| Minimum number of particles | 150 |
| Number of particles per target | 50 |
| Number of birth particles | 100 |
| Probability of false alarm ($p_{FA}$) | $10^{-4}$ |
| Bistatic radar cross section of targets ($\sigma_{RCS}$) | 10 dB |

The Signal-to-Noise ratios (SNRs) and probabilities of detection ($p_D$) for each transmitter-receiver pair in this simulation are given in Figures 17 and 18. The area where the $p_D$ is greater than 0.95 for all the transmitters, and thus the only region in the FoV where the simulation allows particles, is seen in Figure 19. The centers of the gridspaces of the range-variance based grid that is used to determine the ellipse intersections and to distribute birth particles (see Section 5.3.3) are shown in Figure 20. Both Figure 19(b) and Figure 20 provide good illustrations of the geometry of the scenario configuration and antenna gain pattern. The receiver is located in the center of the 160 km $\times$ 160 km field of view (FoV) at location (80 km, 80 km). It is pointed to look at targets over the North Sea, and we assume that its boresight is directed Northwest at an azimuth of 315°.

Figure 21 contains an example of the PHD filter at time $k = 81$. As indicated, the PHD filter has detected two targets. There are actually four targets present in the simulation at that time; however, two of them are in the null of the receiver antenna pattern and are thus ignored, as they are in areas of low $p_D$. Recall that particles are not allowed in areas of low $p_D$ in the current simulation. Figure 22 shows a close-up of the particles surrounding the true targets. Each particle is represented as a $(\dot{x}, \dot{y})$-velocity vector located at the particle's $(x, y)$ position.

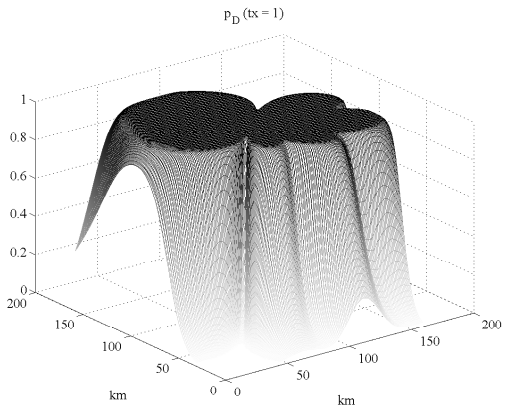



(a) Transmitter 1

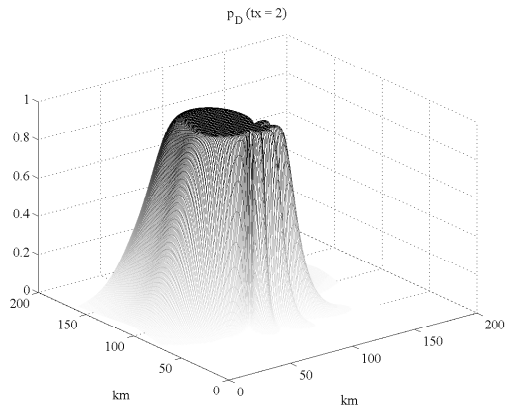
(b) Transmitter 2

(c) Transmitter 3

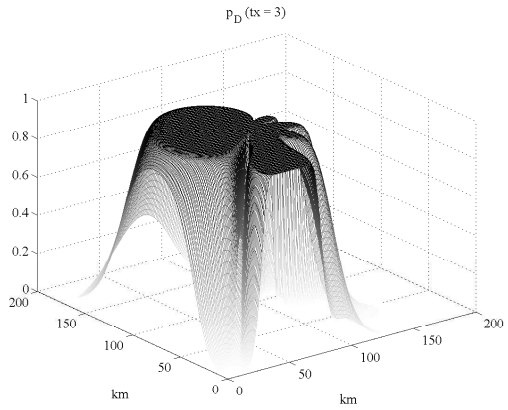
**Figure 17:** The signal to noise ratios of each transmitter/receiver pair in the FoV. The SNRs have been truncated at 10dB so that the areas of low SNR may be seen more clearly.




(a) Transmitter 1

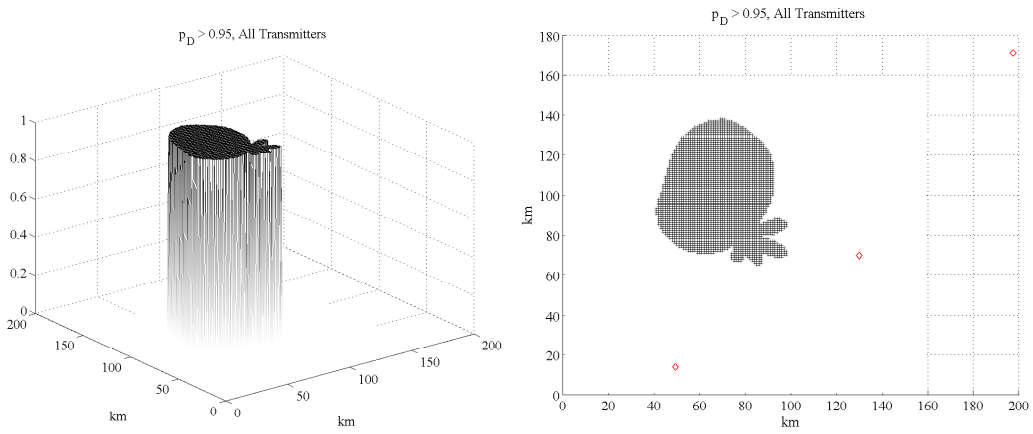



(b) Transmitter 2




(c) Transmitter 3

**Figure 18:** The probabilities of detection of each transmitter/receiver pair in the FoV.




(a) (b)

**Figure 19:** The area in which the probability of detection is greater than 0.95 for all transmitter/receiver pairs. In 19(b), the transmitters are indicated by diamonds, and the receiver is located in the center of the FoV at (80 km, 80 km).

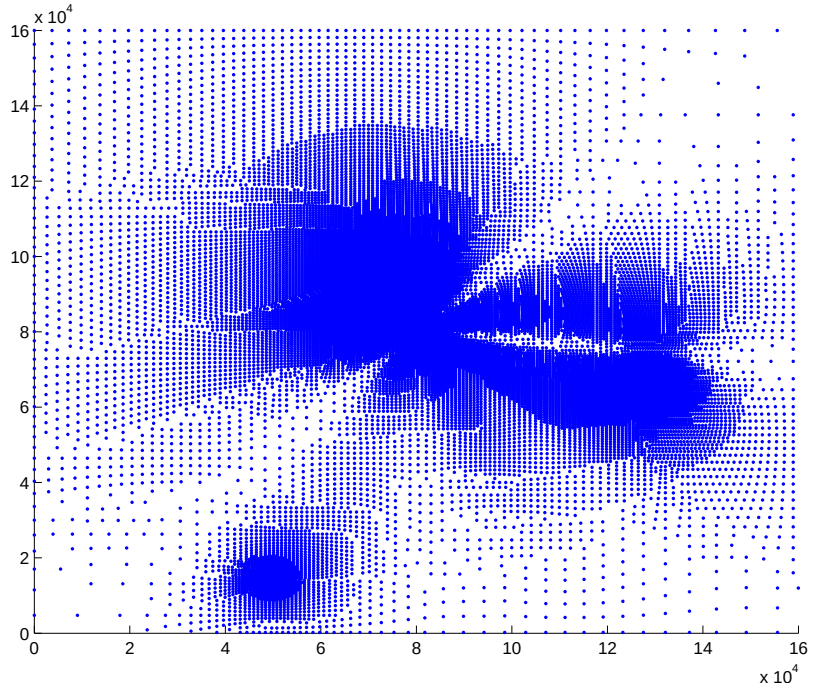



**Figure 20:** Gridpoint locations of the bistatic range-variance based grid computed offline. These gridpoints indicate the centers of the gridspaces used to locate ellipse intersections and distribute birth particles. The gridpspaces are $4\sigma_R \times 4\sigma_R$ and at least 10 meters wide, and they generally overlap each other by $2\sigma_R$. The gridpoint centers are no closer than $1.7\sigma_R$.

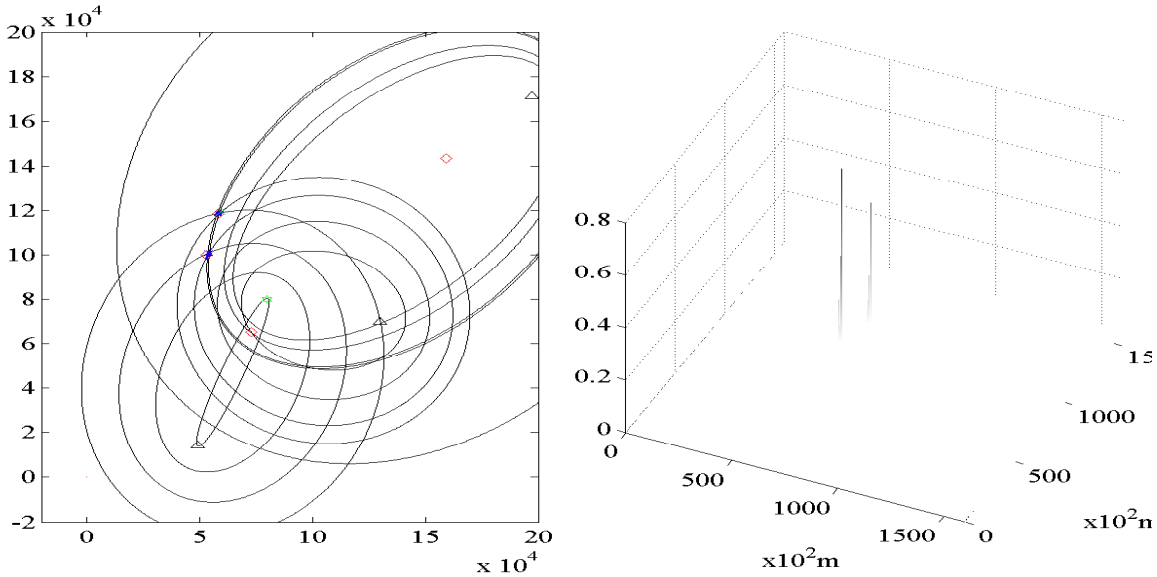



**Figure 21:** The PHD particle filter and range ellipses at time $k = 81$ are shown in the plot on the left. The receiving antenna is represented by the hexagon, and transmitting antennas by the triangles. The diamonds indicate the actual target positions. Each particle of the filter is pictured. The corresponding particle weights are shown on the right. The sum of the weights is 1.999

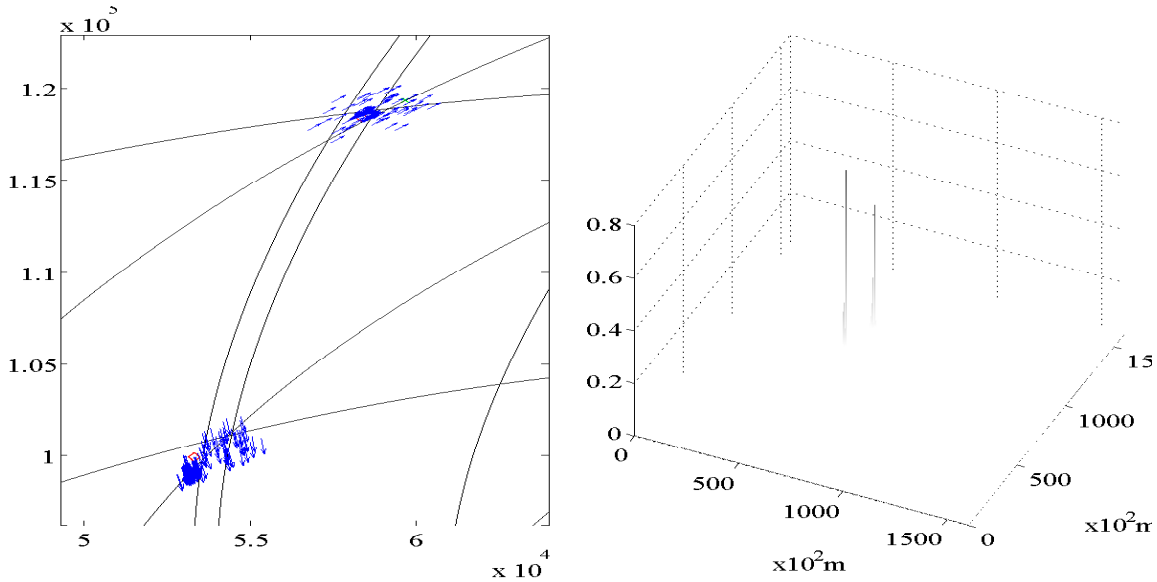



**Figure 22:** A close-up of the PHD particle filter and range ellipses at time $k = 81$

For the two detected targets, Table 14 compares their true states with the estimated states given by the PHD filter.

**Table 14:** Target States at $k = 81$: Actual vs. Estimated

| | Target 1 | | | Target 2 | | |
|---|---|---|---|---|---|---|
| | TRUE | ESTIMATED | ERROR | TRUE | ESTIMATED | ERROR |
| x | $58,354$ m | $59,648$ m | 1294 m | $53,305$ m | $54,174$ m | 869 m |
| y | $118,354$ m | $119,303$ m | 949 m | $99,927$ m | $100,957$ m | 1030 m |
| $\dot{x}$ | 91.92 m/s | 96.59 m/s | 4.67 m/s | 39.07 m/s | 36.09 m/s | -2.98 m/s |
| $\dot{y}$ | 91.92 m/s | 89.58 m/s | -2.34 m/s | -221.58 m/s | -222.74 m/s | -1.16 m/s |

A comparison of the estimated target positions versus the true target $(x, y)$ coordinates are given in Figures 23 and 24. Note that there are five targets present. The targets enter at time steps $k = 1, 15, 22, 30$ and 110. However, only two of them (the ones that enter at $k = 1$ and 22 are ever in regions of high enough $p_D$, so they are the only targets that are detected. The estimated number of targets at each time step is shown in Figure 25. One can see that at $k = 24$, the second target is detected. (Recall that the previous iteration of the PHD filter is at $k = 21$, since we wait until all three transmitters have collected data before running the PHD filter.) The first target moves out of the area of high $p_D$ around $k = 288$, and the second target moves into an area of low $p_D$ between $k = 207$ and $k = 225$. As seen in the figure, the PHD filter correctly estimates the number of targets present, except for when it extracts a ghost target due to false alarms at $k = 93$. The estimated target velocity components are contrasted with the true target velocities in Figures 26 and 27. The errors in position and velocity estimates for the 300 time steps are plotted in Figure 28. Given that the range resolution in the simulation is around 6.7 km, and the Doppler resolution is a little over 3 m/sec, the resulting position and velocity errors are quite reasonable.

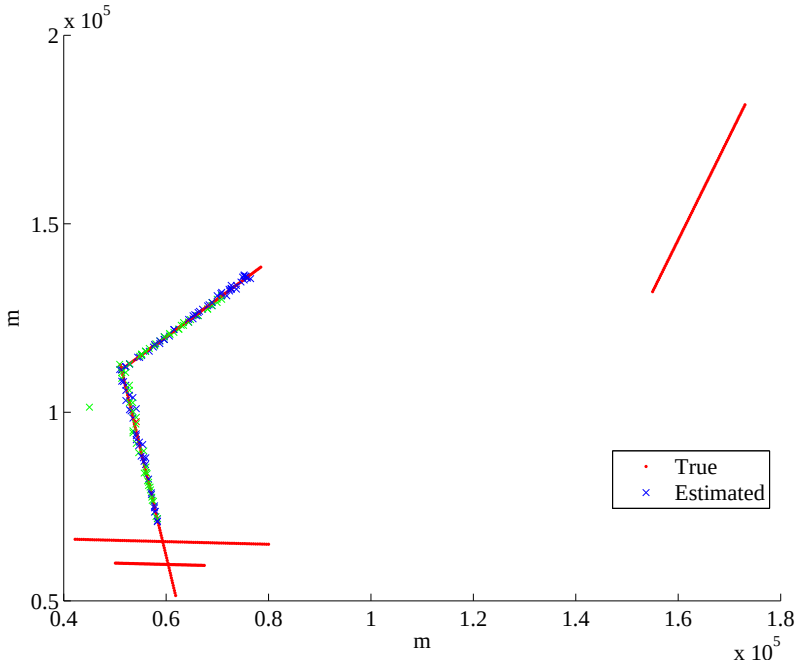



**Figure 23:** Actual vs. estimated target locations over first 300 iterations.

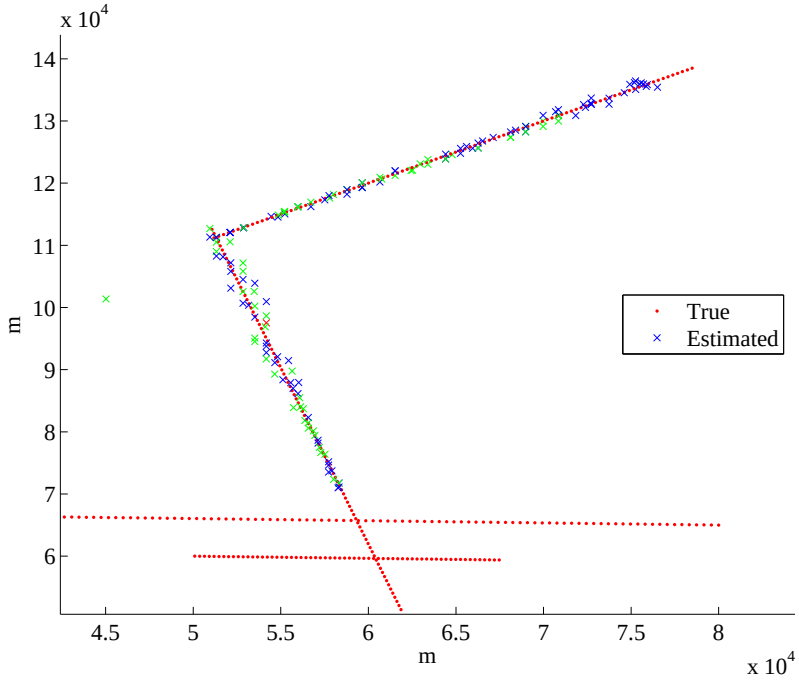



**Figure 24:** Close-up of actual vs. estimated target locations over first 300 iterations.

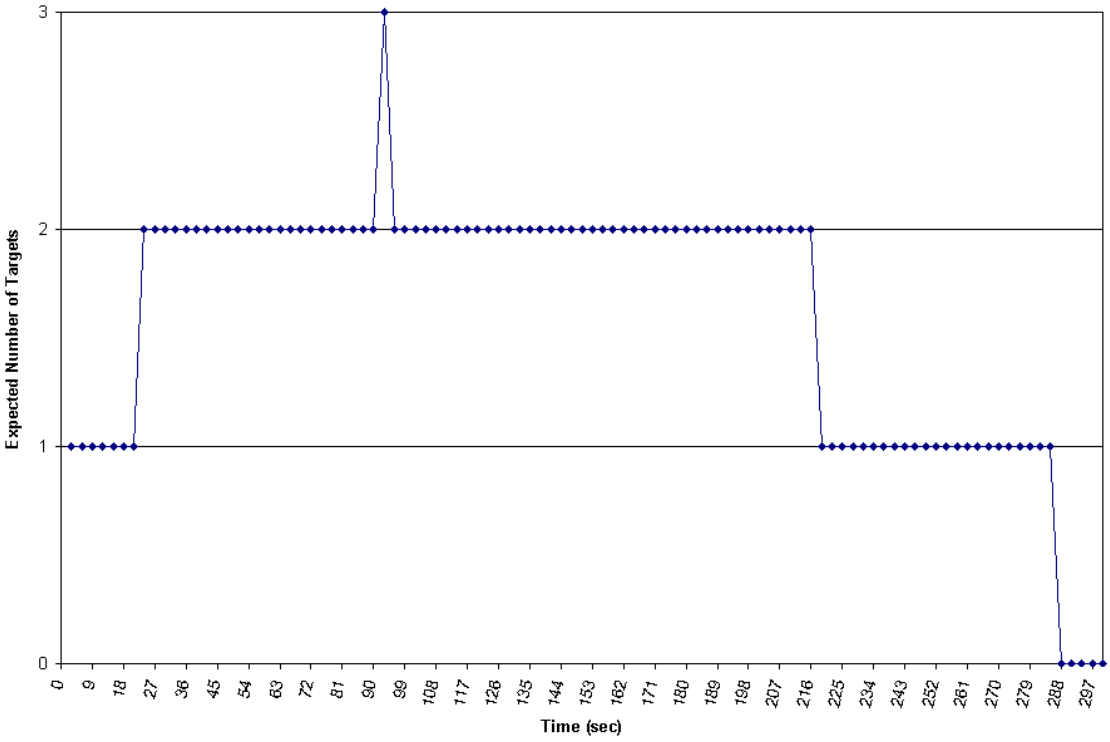



**Figure 25:** Expected number of targets at each time step.

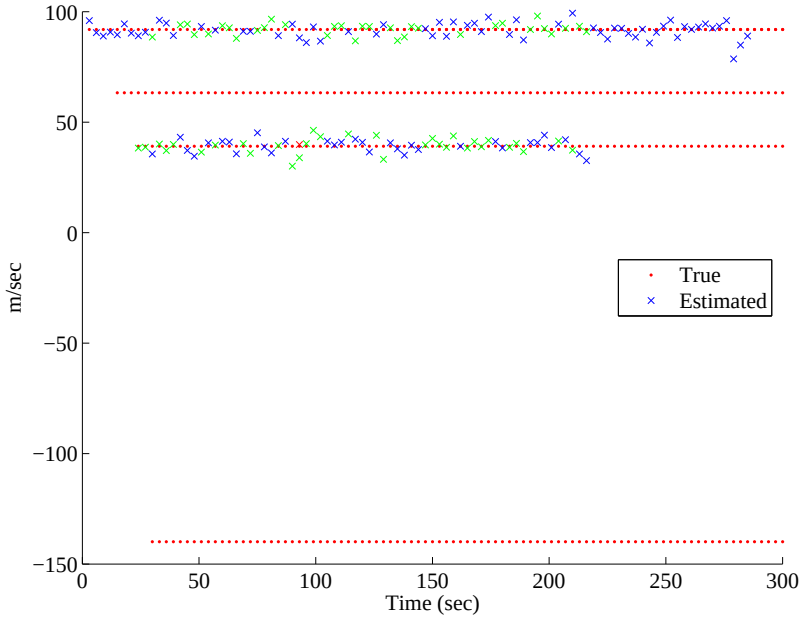



**Figure 26:** Actual vs. estimated target velocities over first 300 iterations.

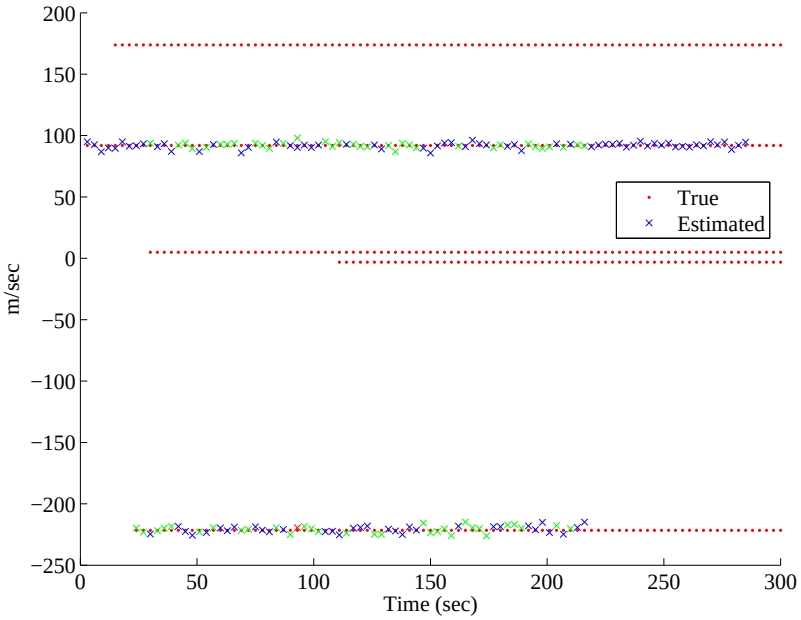



**Figure 27:** Actual vs. estimated target velocities over first 300 iterations.

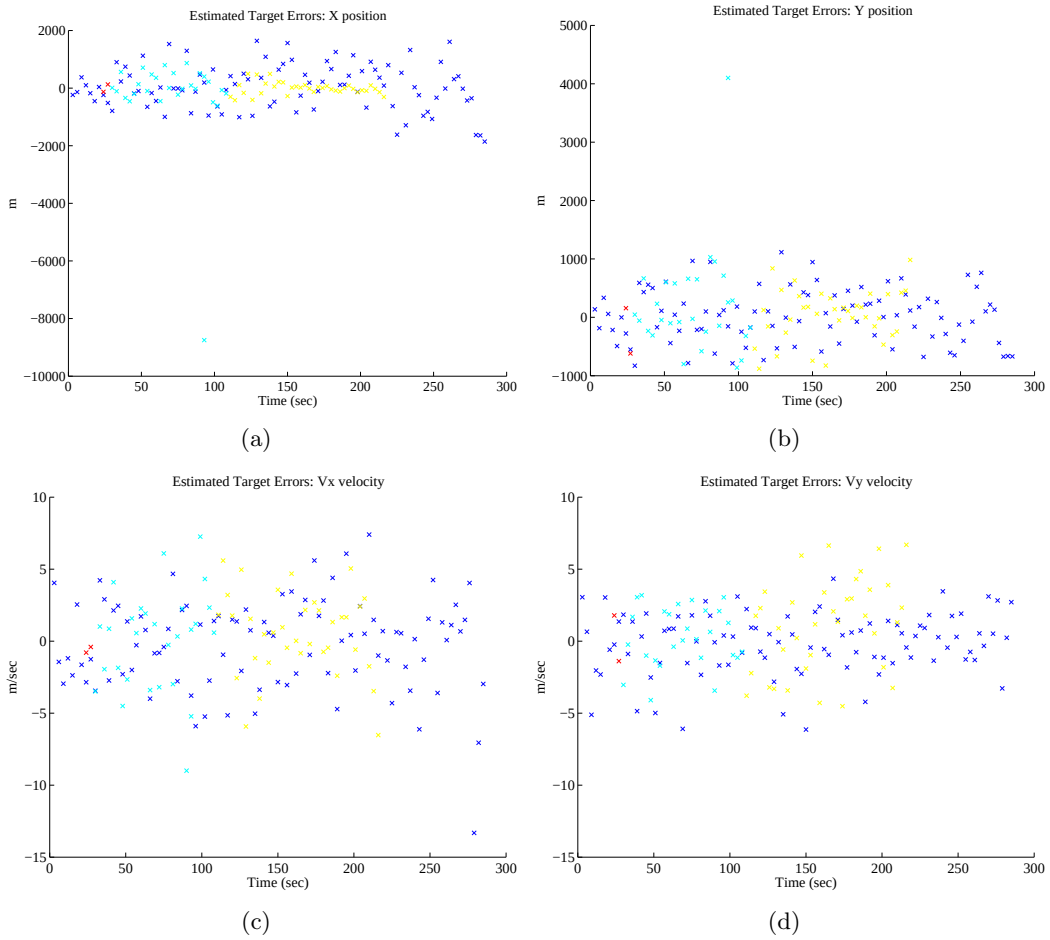



**Figure 28:** Errors in target state estimation by PHD filter. The range resolution is around 6.7 km, and the Doppler resolution is a little over 3 m/sec. Note that in Figure 28(d), there is an outlier at 320 m/sec at $k = 93$ sec, which is not shown.

## 5.5 Conclusions

When the pre-computed $\sigma_R$-spaced grid is used, tracking functionality no longer appears to be dependent on SNR. Furthermore, not only does one not need to increase the number of particles to maintain tracking-ability, as was the case when the range-resolution spaced grid was used, but one can even track targets with fewer particles altogether. In the simulation, the number of particles was reduced by a factor of 10, and this caused no adverse effects in tracking performance.

The major bottleneck in processing performance is, by far, the ellipse-intersection logic. Even if the iterative least-squares technique is used, as described in Section 5.3.5, instead of the grid techniques, an exhaustive combinatorial search is still required to find all of the ellipse intersections. This is computationally feasible when there are few false alarms (on the order of 20-30 per time step), but it becomes quite intractable when the number of false alarms is much greater (such as when $p_{FA} = 10^{-2}$, as described in Section 4.5.3).

Thus, an even smarter birth particle placement method may be desired to allow the filter to run in real-time in the presence of many false alarms. One possible remedy would be to fix the probability of false alarm at a reasonable level and simply settle for a lower probability of target detection. It may also be possible to achieve an immediate speed up in run-time by parallelizing the combinatorial ellipse intersection logic.

It was hoped that the PHD filter could be tested on real data collected from the passive radar at NC3A. However, due to issues in the receiver software, this was not possible. Hopefully in the near future, it will be possible to obtain usable data to run through the PHD filter. In the meantime, we have added further realism, including multipath effects and allowing particles to exist in areas of low SNR, to the simulation. This is discussed in the next chapter.

# CHAPTER VI

# ADDITIONAL IMPROVEMENTS AND ROBUSTNESS TESTS

## *6.1 Introduction*

This chapter presents a variety of features that have been added to the PHD particle-filter simulation presented in Chapter 5. These additions were implemented with the goal of either improving the multitarget-tracking performance of the PHD filter or of increasing the realism of the PHD filter simulation. In Section 6.2, we describe an attempt at injecting explicit target track functionality into the PHD filter. Sections 6.3 and 6.4 present improvements in the realism of the PHD particle-filter simulation, whereby particles were allowed to exist in areas of low signal-to-noise ratio (SNR) and multipath effects were incorporated. An initial implementation of direction of arrival (DOA) observations was incorporated into the PHD filter and is described in Section 6.5. Finally, a target with varying heading and velocity was implemented to test the robustness of the current PHD particle-filter implementation. The results are described in Section 6.6.

## *6.2 Track-ID Augmented Particle Filter Experiment*

The reason the track-ID augmented particle filter appeared to work so poorly was thought to be due to the incorrect use of birth particles. The implementation in Chapter 5 simply distributes birth particles everywhere that the observed bistatic range ellipses and Doppler measurements intersect. This is better than distributing birth particles around the edges of the FoV, as was done in the implementation described in Chapter 4, because it allows us to detect targets that we may have missed entering the FoV, as well as targets which may already be inside the FoV when they first appear, such as airplanes taking-off[1] or those

[1]Ideally, one would always distribute birth particles with the appropriate Doppler values in a region around the airport, since this is a constant source of new targets.

already present when target tracking begins. However, this is not quite the correct use of birth particles, since we do not expect a new target to appear at every ellipse intersection. In fact, we should not place birth particles at those range ellipse and Doppler observation intersections where we already have particles, since we do not expect a new target to appear at the same location as an existing target. If we did, then this should be treated as a spawning target and must be handled differently than a birth target, as specified in the Time Update equation (see Table 6). It was thought that this is the reason why the current implementation of the track-ID augmented particle filter does not work. The birth particles, which have their own track ID, were thought to be interfering with the propagated particles from the previous time step and preventing a real target track from being established.

Thus, the simulation was modified so that birth particles are placed only at intersections where there are currently no propagated particles. In addition, instead of assigning the same track ID to all birth particles, a new track ID is assigned to the group of birth particles at each location where birth particles are placed in the time step. Even though we expect only one new target at each time step (and, hence, we set the weights of the birth particles to sum to one), it is not possible that a target located at an ellipse intersection in one area of the FoV can be the same target located at a different ellipse intersection on the other side of the FoV. Thus, we must assign different track IDs to the birth particles located in different locations.

The reasoning behind the track ID idea, as introduced in Section 5.2.6, was that each of the particle states could be treated as possible target tracks. Even though the particle filter is used to model the probability hypothesis density, it is the particle weights that represent the PHD, while the particle states are the points at which the PHD is sampled. Thus, it was thought that the functionality of the particle states could be expanded to maintain target track information. Sinha *et al.* [71] introduces a tagged particle filter for tracking spawning targets where each track is represented by separate groups of particles. The approach considered here is in a similar vein, except that no grouping of the particles into tracks is performed, and we leave it up to the PHD to propagate the particles correctly.

### 6.2.1 Results

The result of using the track-ID augmented particles with the smarter placement of birth particles can be seen throughout the remainder of this thesis in those figures that display the actual versus the PHD particle-filter estimated target locations and velocities. The track IDs are color-coded, so a change in color of the target-estimate plot-marks indicates a change in the extracted track ID. See, for example, Figures 33-35. Birth particles were excluded from being placed in those gridspaces, or gridspaces adjacent to gridspaces, that already contained particles that had been propagated from the previous time step. The results are not promising. Target tracks were unable to last longer than a few time steps before they were replaced with another track ID number.

The poor performance of the track-ID functionality may indicate two things. First, a better model for the propagated birth particles may be needed, so that the propagated particles track the targets better and effect a longer lasting track. Second, the PHD particle filter may simply be unsuited to maintaining explicit track information. In the track ID method, we are trying to use the particle state to represent a single target track. However, the PHD particle filter is used to model the joint state of the system sampled at single target states. Thus, attempting to use the particles to track single targets may not be possible. For example, a particle that is equidistant from two observations will be weighted correctly in terms of modeling the PHD density. However, what track ID the particle should maintain becomes an issue. The appearances of false alarms and their associated particles may also undermine the accuracy of the track-ID functionality. Essentially, we may be trying to add a feature to the PHD for which it is not designed to handle and which may be better suited for post-processing by some track-forming algorithm.

## *6.3 Particles In Areas of Low SNR*

In the simulation of Chapter 5, particles were restricted to areas where the probability of detection ($p_D$) was greater than 0.95 for all the transmitters. This is a successful strategy to limit the overestimation by the PHD filter of the number of targets present. However, it is an unacceptable limitation, since it considerably restricts the tracking coverage, as seen

in Figure 19(b). When multipath is introduced into the system, as is done in Section 6.4, this restriction becomes further unrealistic.

The restriction of particles to high SNR was originally introduced to avoid the problem described in Section 4.6.1, whereby the number of targets was being considerably overestimated. This was due to the birth particle placement along the edge of the field of view (FoV) and into areas of low SNR. The filter was overestimating the number of targets, since it had to assume that there were targets there which it could not see, because the areas of low SNR are also areas of low $p_D$. However, when the smarter birth particle placement introduced in Section 5.3 is used, the original problem no longer exists. The reason is that the birth particles are now placed at observation locations (namely, the bistatic range ellipse and Doppler observation intersections). The regions into which the birth particles are placed cannot be below the minimum SNR required for the PHD filter to adequately handle, since they are areas in which the PHD filter has already received observations.

Furthermore, the case in which existing particles are allowed to propagate from areas of high SNR into areas of low SNR should also avoid the problem found in Section 4.6.1, since the filter will handle the particles appropriately during the transition. This is seen in the multipath simulations of Section 6.4.2 when the targets cross into regions of low SNR. The original overestimation problem was due not to particles that moved out of areas of high SNR, but to a poor birth model where particles were spontaneously placed in areas of low SNR.

In light of the improved birth particle placement of Section 5.3, the functionality to restrict particles to the FoV, in general, and to areas of high SNR, in particular, was removed. The only place in the simulation where the functionality was kept was in the initial random placement of particles at time $k = 0$.

## *6.4 Multipath Effects*

### 6.4.1 Introduction

The PCL simulation in Chapter 5 proved promising in terms of the effectiveness of the PHD filter for use in multitarget tracking. However, the simulation was an idealized one;

it did not include any multipath effects that would be expected to occur in a real system. Multipath effects occur due to the interaction of the radio waves with the ground and other physical surfaces as they propagate from the transmitting antenna to the receiver. These effects show up in the signal-to-noise ratio and are modeled as the squared propagation factors, $F_R^2$ and $F_T^2$, in the equation for the bistatic radar constant $K$ given in (123). In the bistatic passive radar case, the $F_T^2$ term accounts for the multipath effects in the transmitter to target path, while the $F_R^2$ term accounts for the effects in the target to receiver path.

The propagation factors are computed using the physical characteristics of the NC3A passive radar described in [28]. A three-dimensional flat-earth model is used, and targets are simulated to fly at an altitude of 7315 m (approx. 24,000 ft) above ground level. The vertically-polarized receiver is located at a height of 20m, and the transmitters are all assumed to be located at a height of 375 m above ground. A depiction of the scenario appears in Figure 29. This new 3D flat-earth model is incorporated into the simulation, so that the bistatic range and Doppler observations take the target heights into account. However, the tracker is not modified to take target altitude into account. That is, no $z$ or $\dot{z}$ parameters are added to the particle states. This results in a slight bias in the observed target locations relative to the true target locations, as evident in Figure 34. As the goal of the research is to present an initial evaluation of the performance of the PHD filter, we begrudgingly leave in this bias, since it does not affect our evaluation of the PHD and to remove it would require a considerable amount of additional computation. Indeed, the bistatic range ellipses would become ellipsoids, and the offline grid, or iterative least-squares, technique for placing birth particles would need to be expanded to three dimensions.[2]

Since the squared propagation factors $F_T^2$ and $F_R^2$ are derived in a similar fashion, the generic term $F^2$ will be used to represent both in the following computations. It will be indicated when necessary where the calculations differ for the two propagation paths. Using the equations and plots given by Barton in Section 6.2 of [3], the squared propagation factor

[2]Furthermore, it is difficult to obtain good altitude information on low-flying targets due to increased geometric dilution of precision (GDOP), which is a measure of the inaccuracy in the radar system's ability to locate targets due to the poor geometries involved. Because of this, some surveillance radars are used to track targets in two dimensions only.

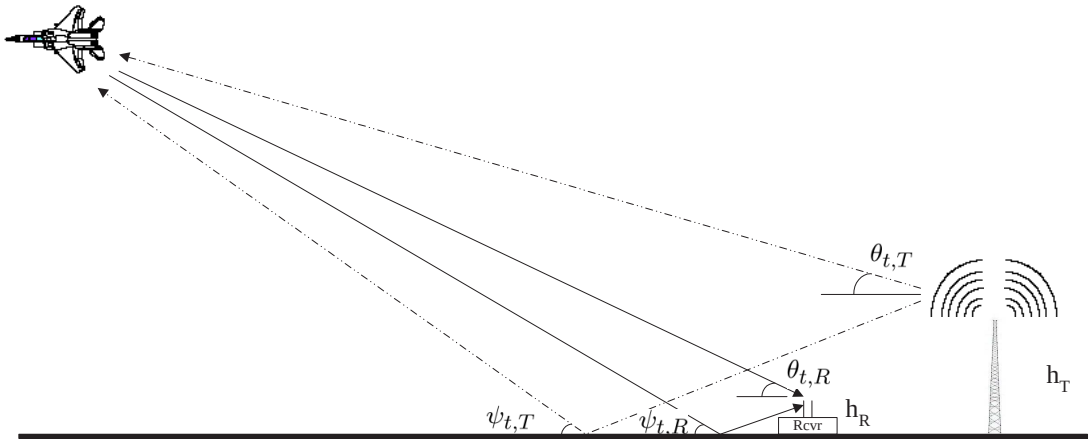



**Figure 29:** A diagram of the multipath between a transmitter and the receiver. The dashed lines are the paths of the radio signal from the transmitting antenna to the target and determine $F_T^2$, while the solid lines are the paths of the echo signal from the target to the receiver that determine $F_R^2$. The elevation angles of the target relative to the transmitter and receiver are indicated by $\theta_{t,T}$ and $\theta_{t,R}$, respectively. The grazing angles are $\psi_{t,T}$ and $\psi_{t,R}$, and the heights of the antennas are $h_T$ and $h_R$ for the transmitter and receiver, respectively.

of a path is found to be:

$$F^2 = 1 + \rho^2 + 2\rho \cos\left(\phi + 2\pi \frac{\theta_t}{\theta_n}\right), \tag{150}$$

where

$$\theta_n = \frac{\lambda}{2h}, \tag{151}$$

where $\lambda$ is the wavelength of the propagating wave, $\theta_t$ is the elevation angle of the target relative to the antenna, and $h$ is either the height $h_R$ of the receiver, in the case of $F_R^2$, or the height $h_T$ of the transmitter, in the case of $F_T^2$. This assumes that $h_T >> h_R$. The phase shift term, $\phi$, is taken to be 180°, given the wavelengths and grazing angles present in the simulation. The surface reflection coefficient, $\rho$, is defined to be:

$$\rho = \rho_0 \rho_s \rho_v, \tag{152}$$

where $\rho_0$ is the Fresnel reflection coefficient of the surface, $\rho_s$ is the specular scattering coefficient of a rough surface, and $\rho_v$ is the vegetative absorption coefficient. The Fresnel reflection coefficient depends on the grazing angle $\psi$ and phase shift $\phi$ of the reflecting wave, as well as on the complex dielectric constant $\epsilon_c$ of the surface material. For vertical

polarization, it is given by the following expression:

$$\rho_0 e^{-j\phi} = \frac{\epsilon_c \sin\psi - \sqrt{\epsilon_c - \cos^2\psi}}{\epsilon_c \sin\psi + \sqrt{\epsilon_c - \cos^2\psi}}, \tag{153}$$

where

$$\epsilon_c = \epsilon_r - j60\lambda\sigma_e, \tag{154}$$

where $\epsilon_r$ is the relative dielectric constant, and $\sigma_e$ is the conductivity, of the surface. Since the NC3A radar is looking out over the sea, we assume that the reflection surface is that of salt water, and hence set $\epsilon_r = 75$ and $\sigma_e = 5$ mho/m. We also assume that the sea water is not a rough surface and that $\rho_s \approx 1$. We assume also that there is no vegetation with a thickness greater than a wavelength present everywhere, and so we assume that $\rho_v \approx 1$.

Thus, as noted in [3], the propagation factor $F$ varies cyclically in target elevation angle with a period of $\theta_n$, where

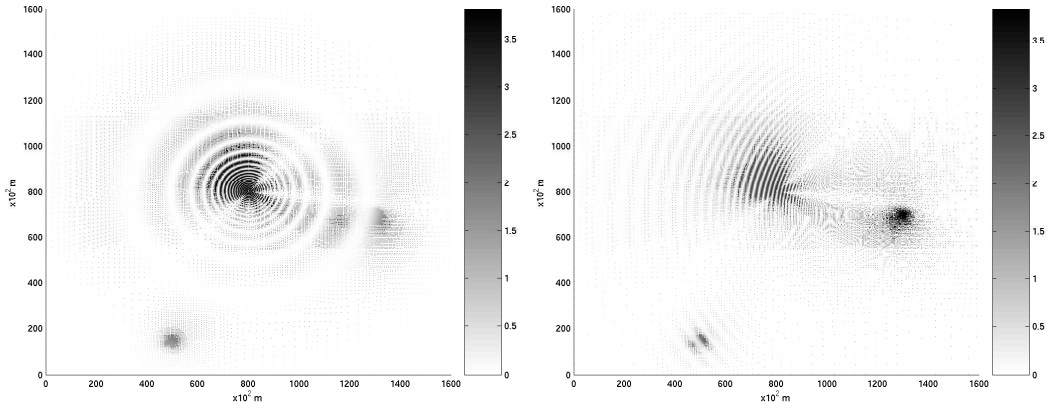
$$F_{max} = 1 + \rho, \text{ and} \tag{155}$$

$$F_{min} = 1 - \rho. \tag{156}$$

The magnitudes of $F_R^2$ and $F_T^2$ are shown in Figure 30 for each of the transmitters used in the NC3A simulation. Note that $F_R^2$ depends on the transmitter only to determine which $\lambda$ to use in computing $F_R^2$. Thus, $F_R^2$ does not vary much based on which transmitter is used. The plots of $F_T^2$ for each transmitter do vary significantly, however. Also, note that since the height of the receiver is less than the height of the transmitters, the rate of the cyclic variation of $F_R^2$ is slower than that of $F_T^2$, as evidenced by the rings in Figure 30. Figure 31 contains the two-way power ratio, $F_T^2 F_R^2$, used to calculate the SNR, via (122), for each of the transmitters involved in the NC3A simulation. In this figure, one can see the rings due to both $F_R$ and $F_T$. The minimum and maximum values of the magnitude of the two-way power ratio are given in Table 15.

### 6.4.2 Simulation Results

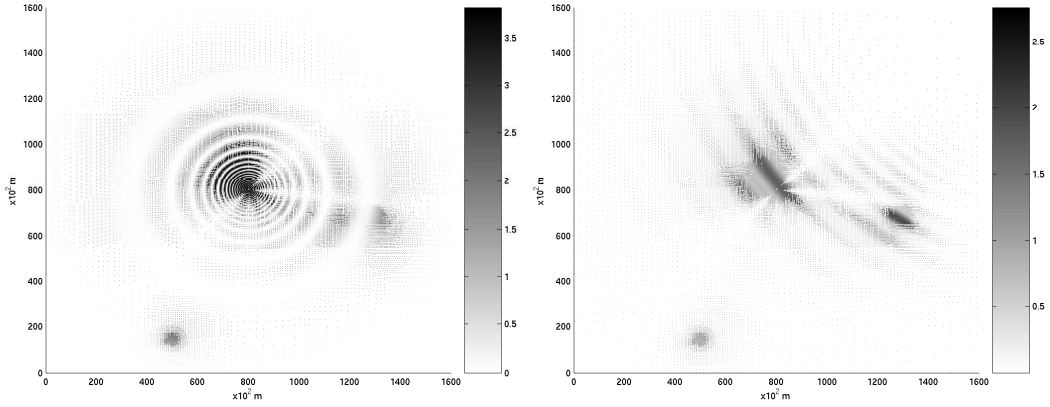
Having incorporated the multipath model into our simulation observations, the PHD filter is now tested for its robustness to slowly fluctuating reflections from the target and the



(a) Transmitter 1, $F_R^2$



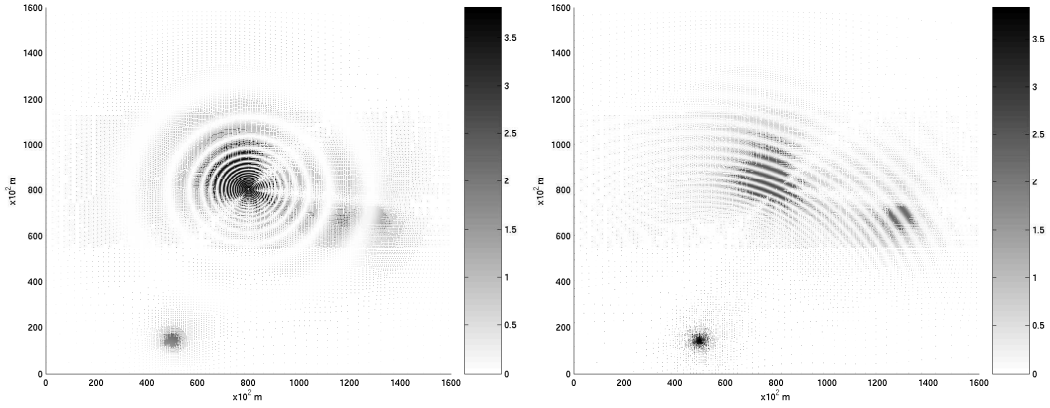
(b) Transmitter 1, $F_T^2$



(c) Transmitter 2, $F_R^2$



(d) Transmitter 2, $F_T^2$

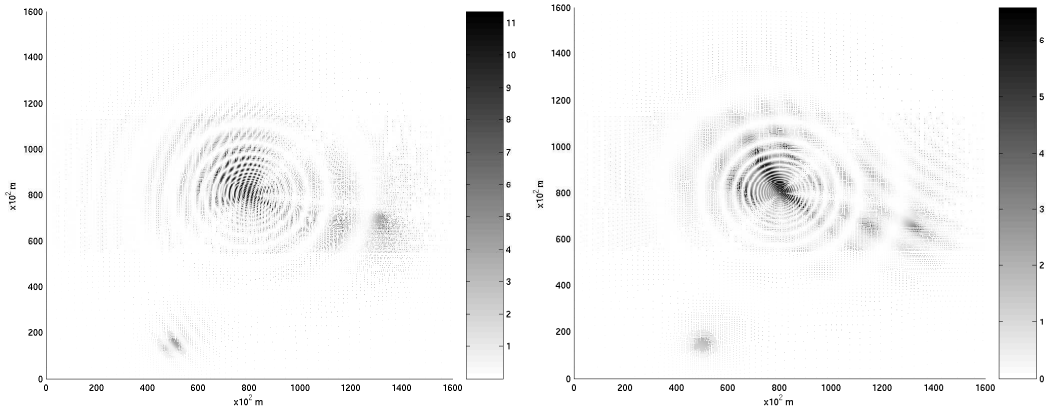


(e) Transmitter 3, $F_R^2$



(f) Transmitter 3, $F_T^2$

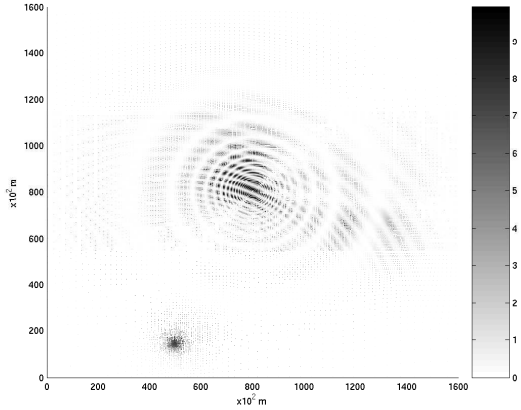
**Figure 30:** The magnitudes of the squared propagation factors, $F_R^2$ and $F_T^2$, used in the NC3A simulation, for a target located at an altitude of 7315 m. The values plotted are those at the centerpoints of the gridspaces in the range-variance based grid.



(a) Transmitter 1, $F_R^2 F_T^2$



(b) Transmitter 2, $F_R^2 F_T^2$



(c) Transmitter 3, $F_R^2 F_T^2$

**Figure 31:** The magnitudes of the two-way power ratio, $F_R^2 F_T^2$, for each of the transmitters used in the NC3A simulation, for a target located at an altitude of 7315 m. The values plotted are those at the centerpoints of the gridspaces in the range-variance based grid.

**Table 15:** Minimum and Maximum Magnitudes of Two-Way, Multipath Power Ratio

| | Minimum $\lvert F_T^2 F_R^2 \rvert$ | Maximum $\lvert F_T^2 F_R^2 \rvert$ |
|---|---|---|
| Transmitter 1 | 0.0006 | 11.35 |
| Transmitter 2 | 0.0031 | 6.58 |
| Transmitter 3 | 0.0011 | 9.93 |

resulting changes in the probability of detection ($p_D$). That is, we want to determine if the PHD filter can still track the targets with range and Doppler observations, as it did in Chapter 5, using the offline grid that was generated without taking $F_R$ and $F_T$ into account, even though the observations now incorporate the multipath effects. For illustration purposes, the simulations in this section consist of only the two targets that were successfully tracked in Chapter 5. Target 1 is present at time $k = 1$ and moves at a speed of 130 m/sec at a heading of 45° in azimuth. Target 2 enters at time $k = 22$ and travels at 225 m/sec with an azimuthal heading of 170°. The values of the SNR and $p_D$ for the two targets throughout the simulation are displayed in Figures 37 and 38.

The performance of the multitarget tracker in estimating the number of targets is shown in Figure 32. The PHD filter exhibits much overestimation, and the occasional underestimation, of target number. Figures 33 and 34 compare the real target locations to those found by the PHD filter and peak-extraction algorithm, and Figure 35 compares the velocities found to the true target velocities. The errors in the estimated target positions and velocities are displayed in Figure 36. Note that these errors are slightly greater than those of Section 5.4. This is due somewhat to the bias introduced by the three-dimensional flat-earth model, as described in Section 6.4.1, as well as to the incorrect tracking of Target 2 in the later stages of the simulation. Note how the estimated location of Target 2 starts to veer off of the true path of the target before entering an area of low SNR for all its transmitters.

As seen in the figures, the two targets were dropped by the range and velocity tracker upon entering regions of low SNR, and then were detected again upon re-entering regions of high SNR. Our explanation for this behavior is that, in the regions where the targets

are dropped, the PHD filter assumes that the targets are in areas of high SNR. Yet, due to multipath effects, the targets are really in regions of low SNR. Thus, the PHD filter ends up having missed observations in an area of assumed high $p_D$, and so the targets are dropped. This is the issue with which Erdinc *et al.* [19] are concerned, in which they claim that the PHD filter drops targets abnormally quickly, relative to a Markov chain model, due to the $(1 - p_D)$ term in the Date Update equation (see (93)). This issue requires further study by the PHD target-tracking community, in general.

Note that a ghost target is detected and followed when the PHD filter loses track of Target 1.

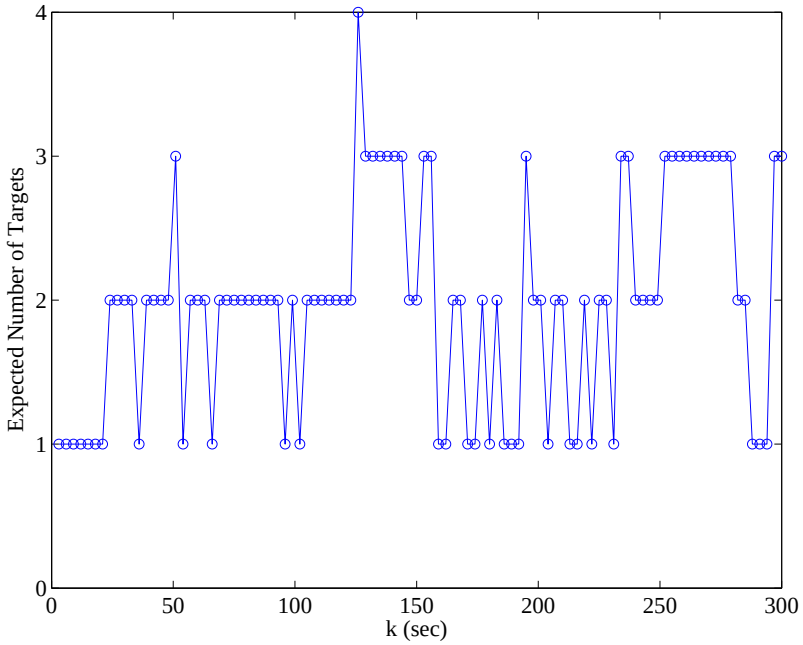



**Figure 32:** Expected number of targets at each time step in the two-target simulation with multipath effects.

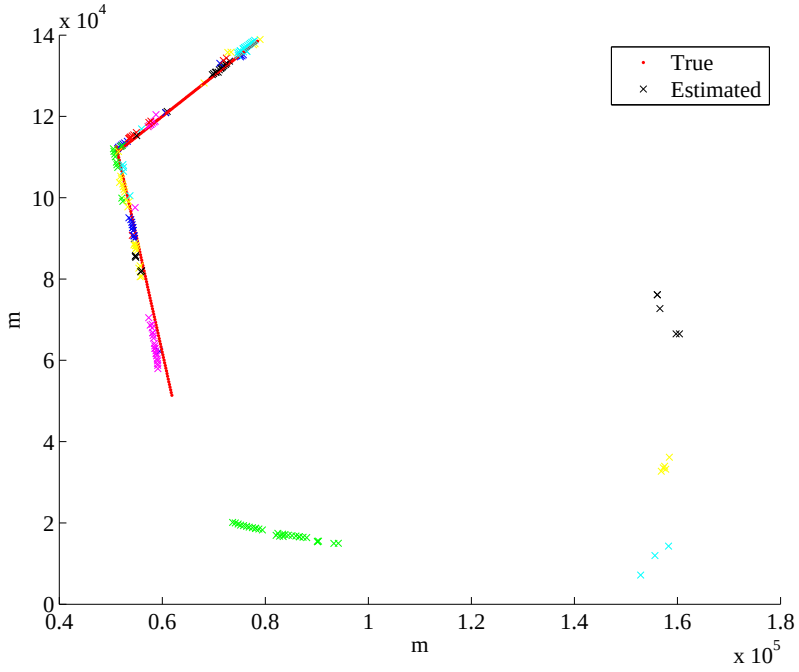



**Figure 33:** Actual vs. estimated target locations over the first 300 iterations of the two-target simulation with multipath effects. Adjacent estimated target locations with the same color have the same track ID.

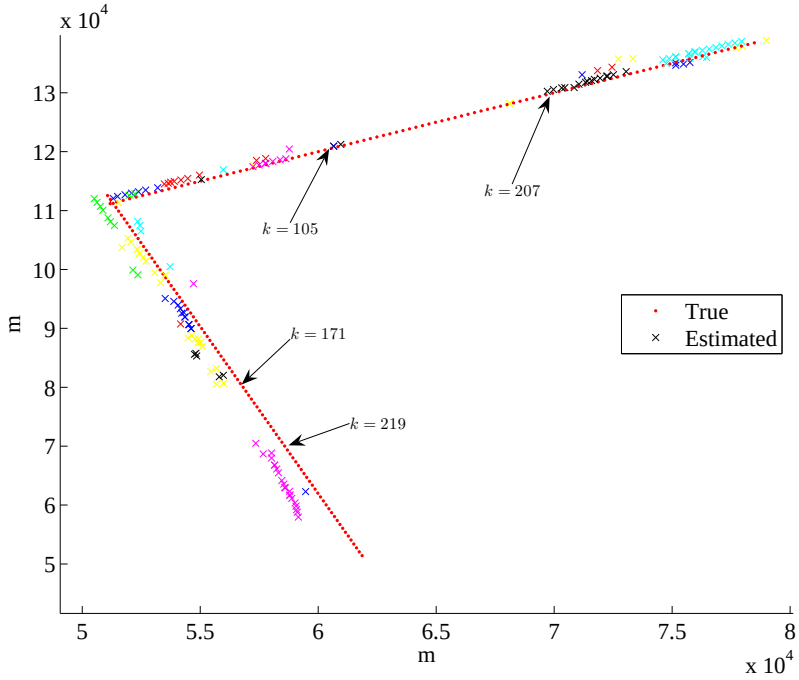
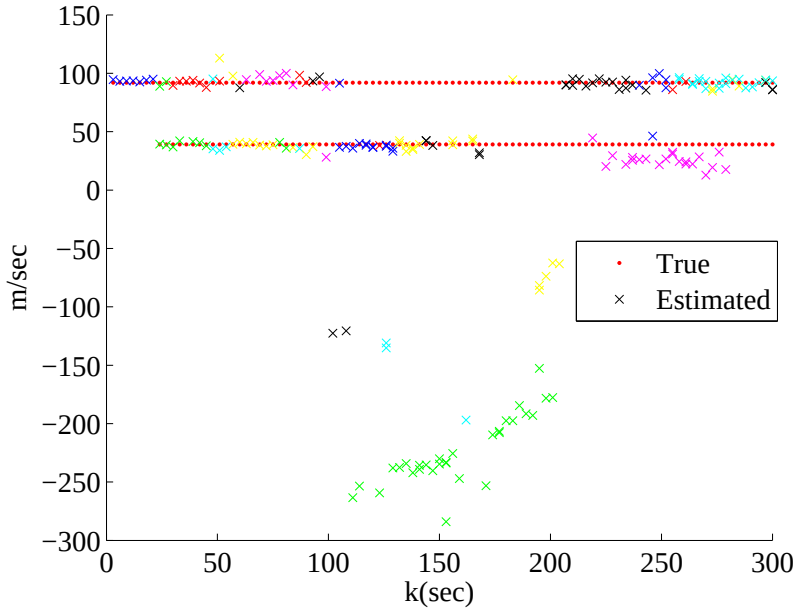
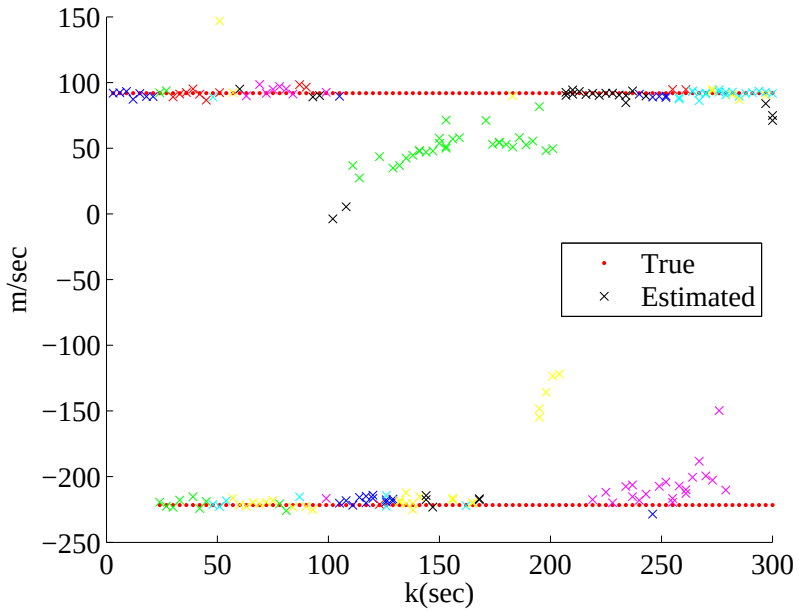



**Figure 34:** Close-up of actual vs. estimated target locations over the first 300 iterations of the two-target simulation with multipath effects. Adjacent estimated target locations with the same color have the same track ID. The time step ($k$) in the simulation is indicated for a few points of interest.




(a) $\dot{x}$




(b) $\dot{y}$

**Figure 35:** Actual vs. estimated target velocities over the first 300 iterations of the two-target simulation with multipath effects. Adjacent estimated target locations with the same color have the same track ID.

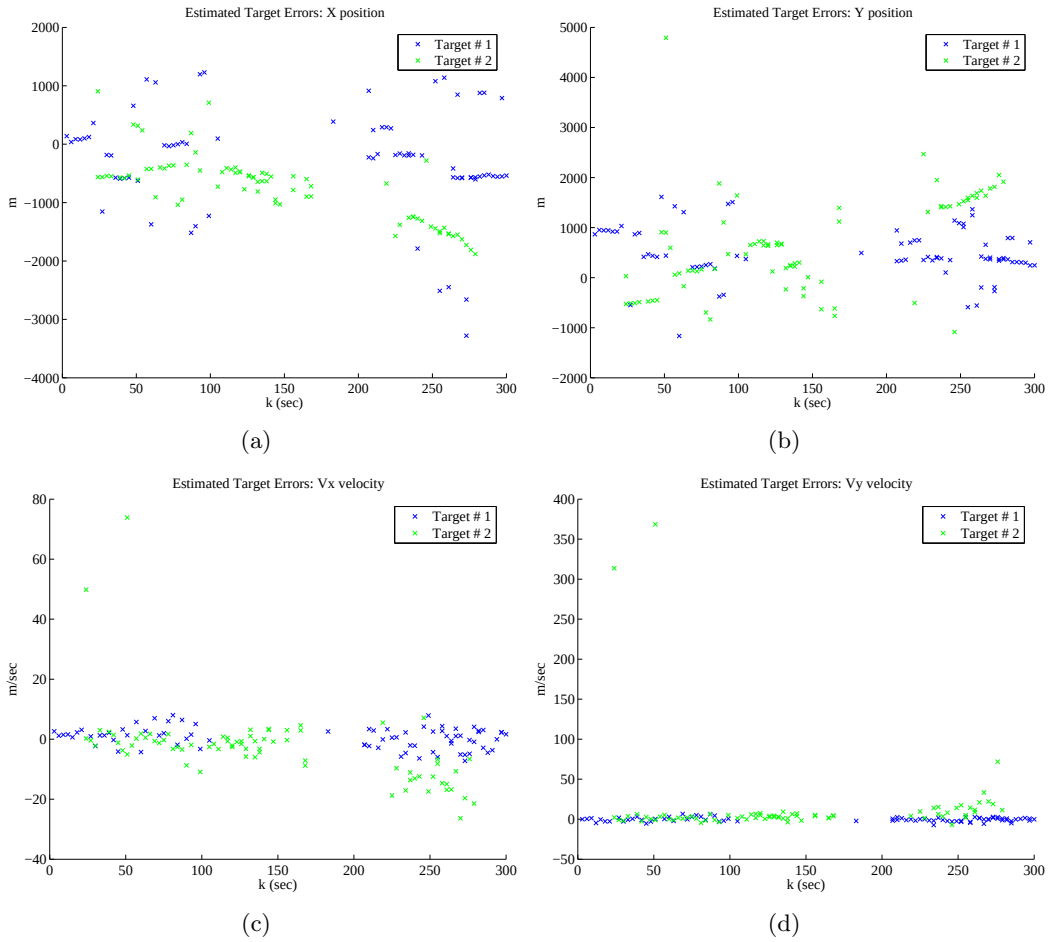
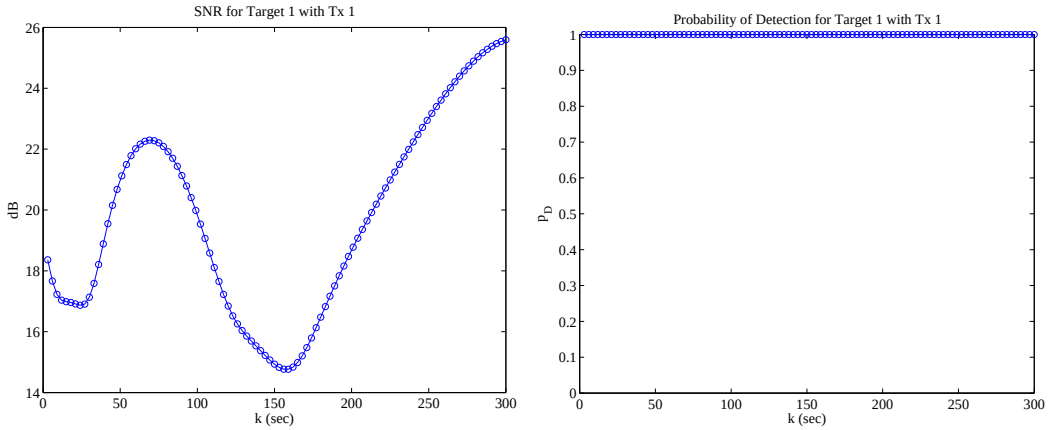


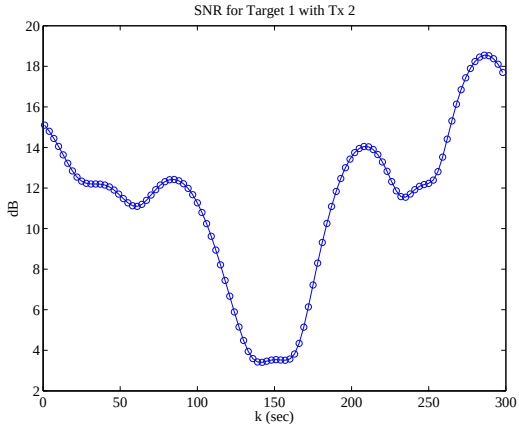

**Figure 36:** Errors in target state estimation by the PHD filter in the two-target simulation with multipath effects. The range resolution is around 6.7 km, and the Doppler resolution is a little over 3 m/sec.



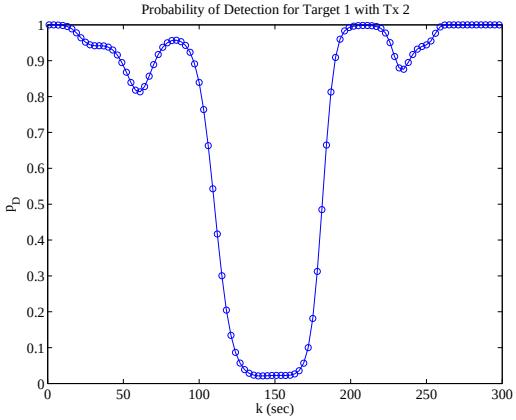

(a) Transmitter 1, SNR



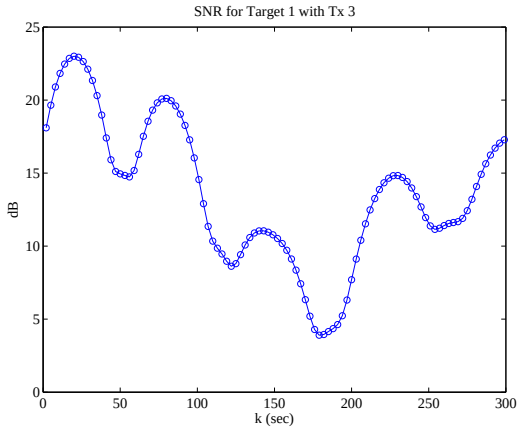

(b) Transmitter 1, $p_D$



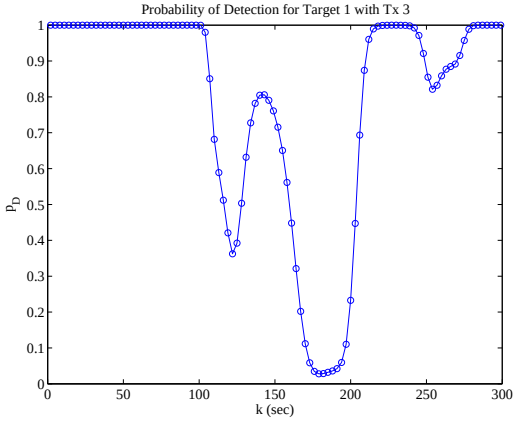

(c) Transmitter 2, SNR




(d) Transmitter 2, $p_D$

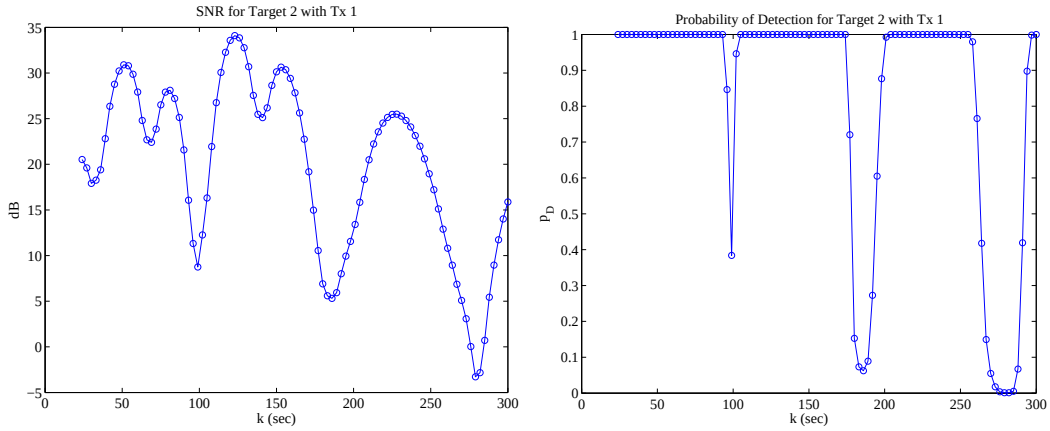



(e) Transmitter 3, SNR




(f) Transmitter 3, $p_D$

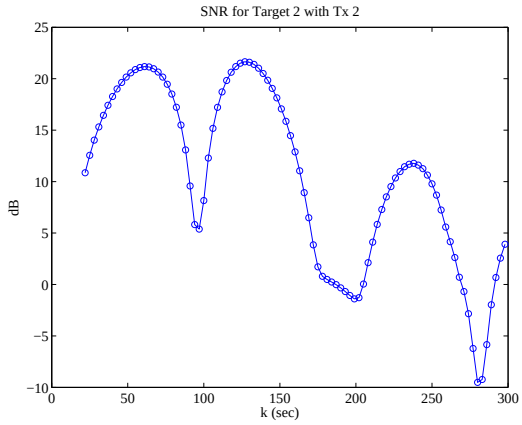
**Figure 37:** The signal to noise ratios and probabilities of detection of Target 1 for each transmitter with multipath effects present. The circles indicate the actual values sampled. The lines are drawn in as a visual aid.



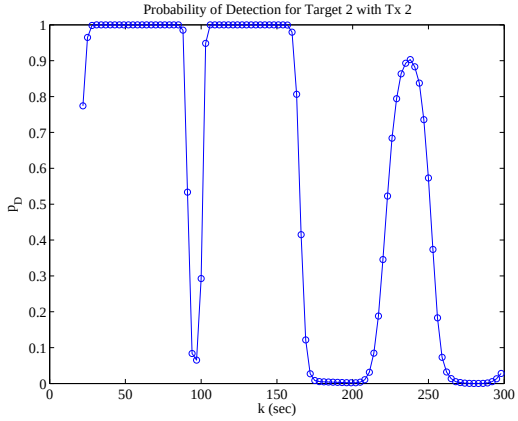

(a) Transmitter 1, SNR



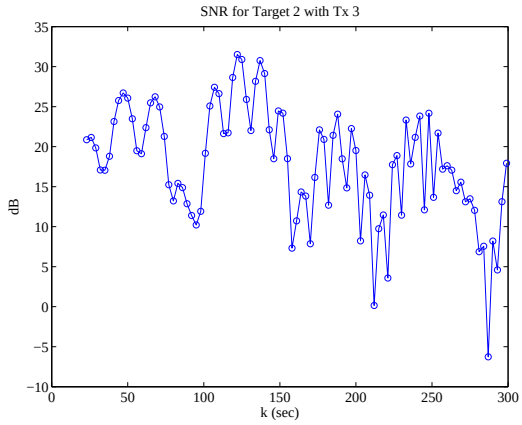

(b) Transmitter 1, $p_D$



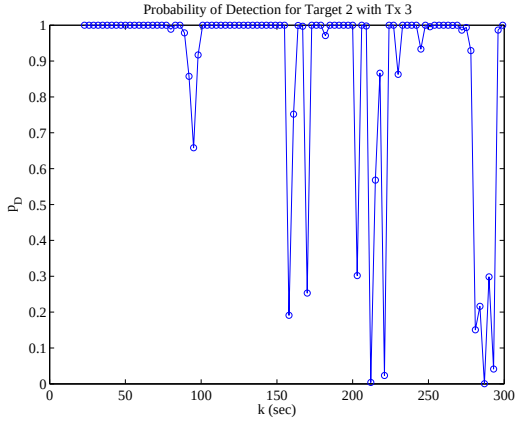

(c) Transmitter 2, SNR




(d) Transmitter 2, $p_D$




(e) Transmitter 3, SNR




(f) Transmitter 3, $p_D$

**Figure 38:** The signal to noise ratios and probabilities of detection of Target 2 for each transmitter with multipath effects present. The circles indicate the actual values sampled. The lines are drawn in as a visual aid.

Additional simulation experiments were performed to gauge the effect of multipath on the PHD filter. Brief descriptions of each experiment and the corresponding results are given below.

**Experiment 1** To obtain a baseline for the performance of the PHD filter, the true $p_D$ and SNR values were used in the Data Update step of the PHD filter. This assumes that the PHD filter knows the true altitude of the target. The result, as seen in Figures 39-42, was similar to the initial multipath, except that Target 1 was almost always detected. Only Target 2 was dropped in the area of low SNR. This experiment also contained more overestimation of target number than in the original multipath simulation, where the PHD filter did not know the true $p_D$ and SNR values.

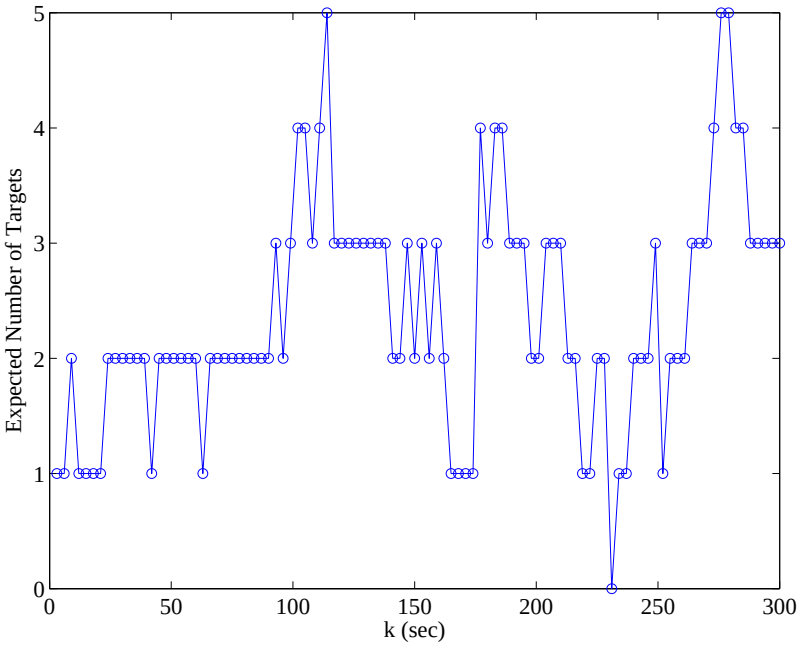



**Figure 39:** Expected number of targets at each time step in Experiment 1 with multipath effects, where the PHD filter knows the $p_D$ and SNR truth.

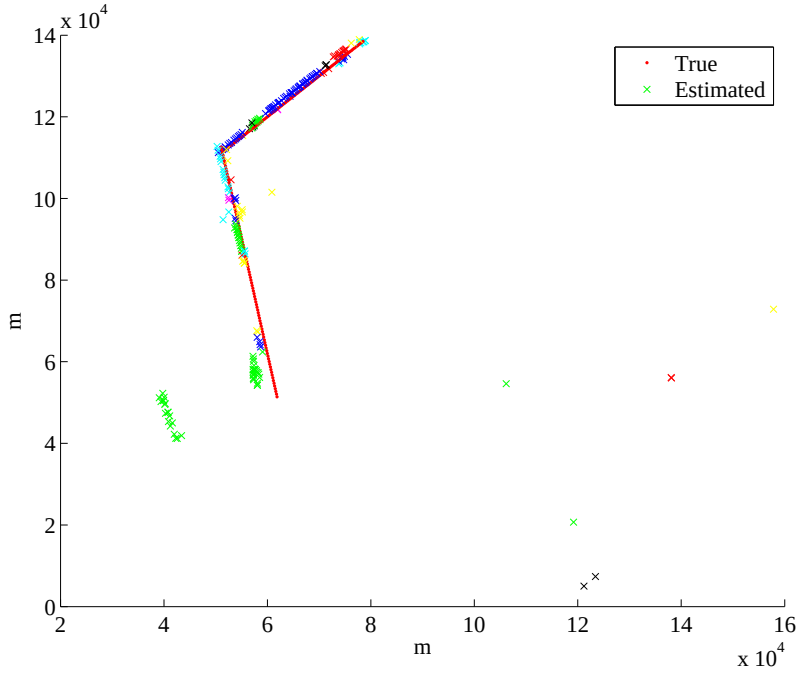
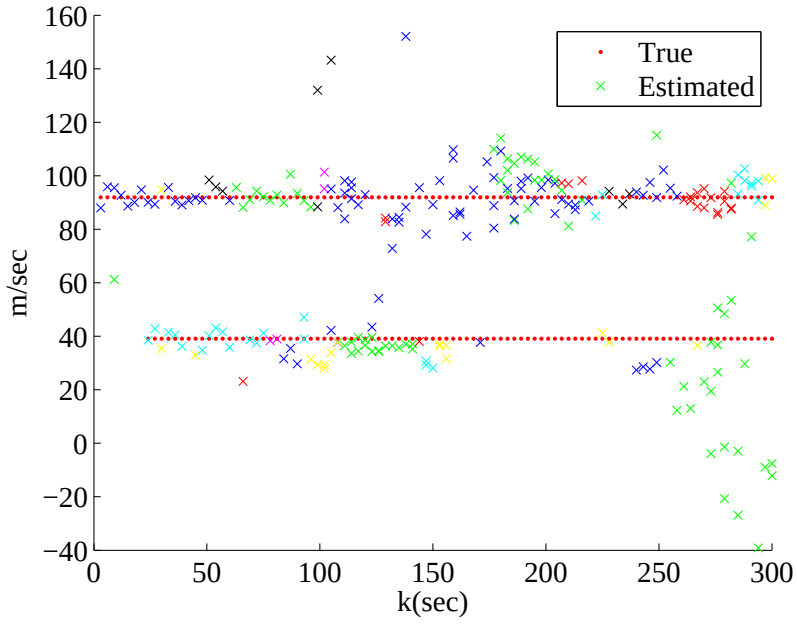
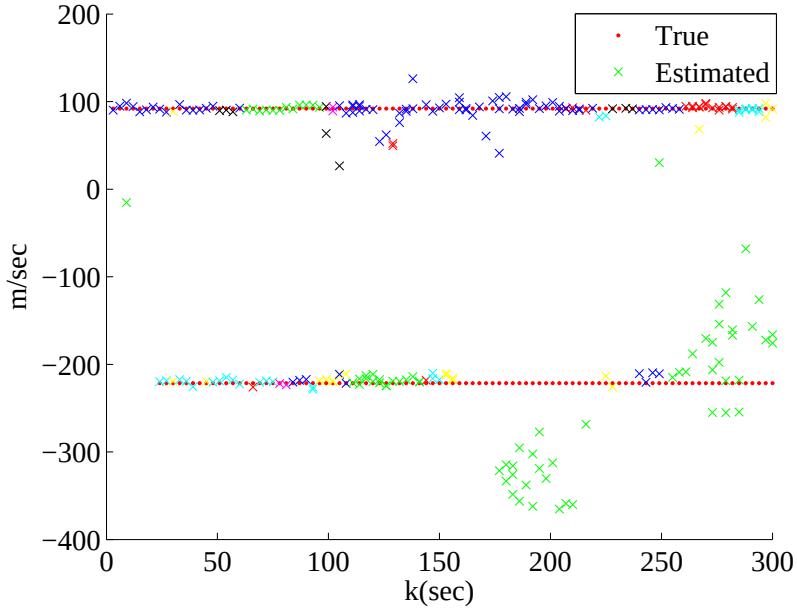



**Figure 40:** Actual vs. estimated target locations over the first 300 iterations in Experiment 1 with multipath effects, where the PHD filter knows the $p_D$ and SNR truth.




(a) $\dot{x}$




(b) $\dot{y}$

**Figure 41:** Actual vs. estimated target velocities over the first 300 iterations in Experiment 1 with multipath effects, where the PHD filter knows the $p_D$ and SNR truth.

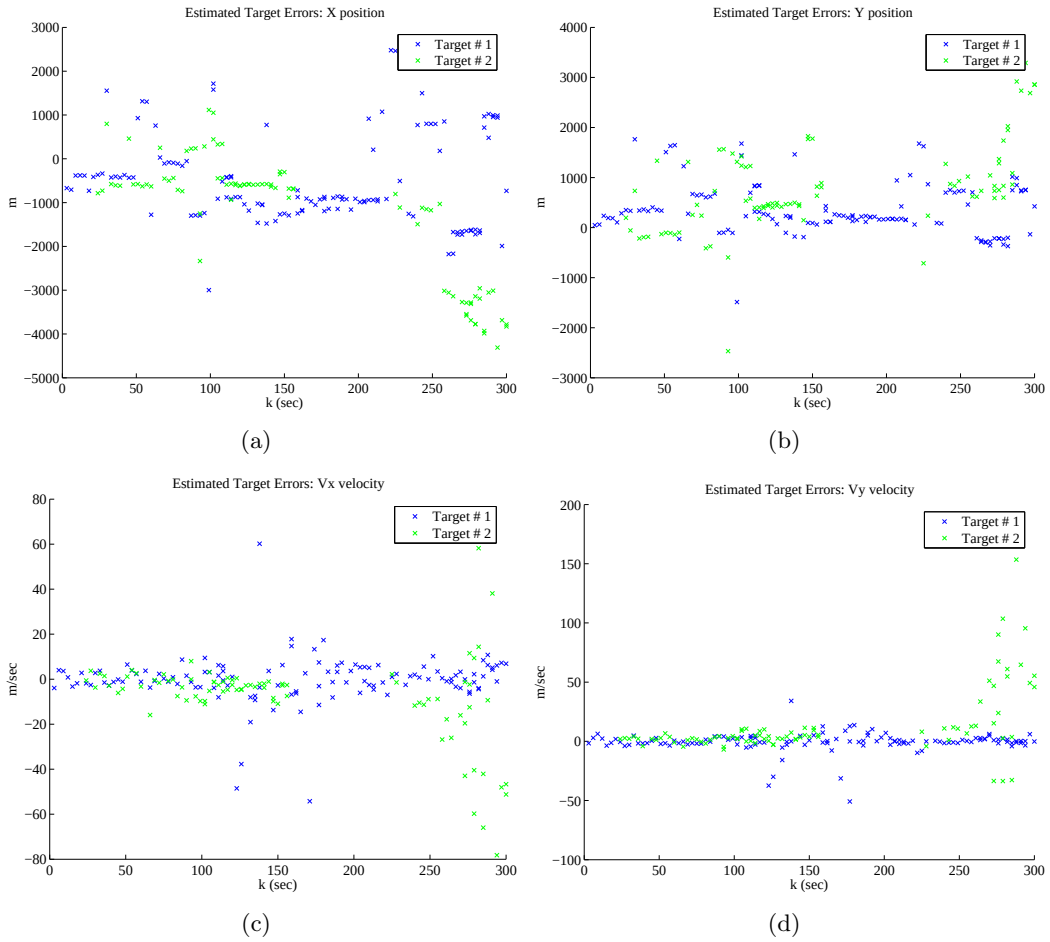



**Figure 42:** Errors in target state estimation by the PHD filter in Experiment 1 with multipath effects, where the PHD filter knows the $p_D$ and SNR truth. The range resolution is around 6.7 km, and the Doppler resolution is a little over 3 m/sec.

**Experiment 2** Instead of using the true $p_D$ and SNR in the Data Update step of the PHD filter, the lowest possible $p_D$ values were used by assuming the worst-case multipath power ratio, i.e., $\min(F_R^2 F_T^2)$, for the whole FoV. The result, as seen in Figures 43 and 44, was that the number of targets was extremely overestimated by the PHD filter. One possible reason for this is that the range likelihood used by the filter was too broad. Since the SNR was too low, the range variance used to compute the likelihood was too large. Thus, the filter considered too many ellipses to be intersections, even if they were separated by large distances.

This simulation was also run with false-alarm suppression to determine whether the extreme target number overestimation would still occur if no false alarms were present. The false alarm parameters were kept as is, so the PHD filter still expected false alarms; however, no false alarms were allowed to occur. This would lead us to expect a slight underestimation by the PHD filter of expected target number. Nevertheless, the results shown in Figures 45 and 46 illustrate that the overestimation still occurs in the absence of false alarms.

Thus, where the $p_D$ used by the PHD filter is smaller than the real $p_D$ in the FoV, the overestimation may be the same overestimation problem seen when birth particles were spontaneously placed in areas of low $p_D$ (see Sections 4.6.1, 5.2.7, and 6.3). Because the PHD filter assumes that the $p_D$ is low, it believes that there are targets present which it cannot observe. Yet, because the actual $p_D$ is higher, the PHD filter does receive observations from the targets. This appears to have a feedback effect in the PHD filter and causes considerable overestimation of the number of targets present.

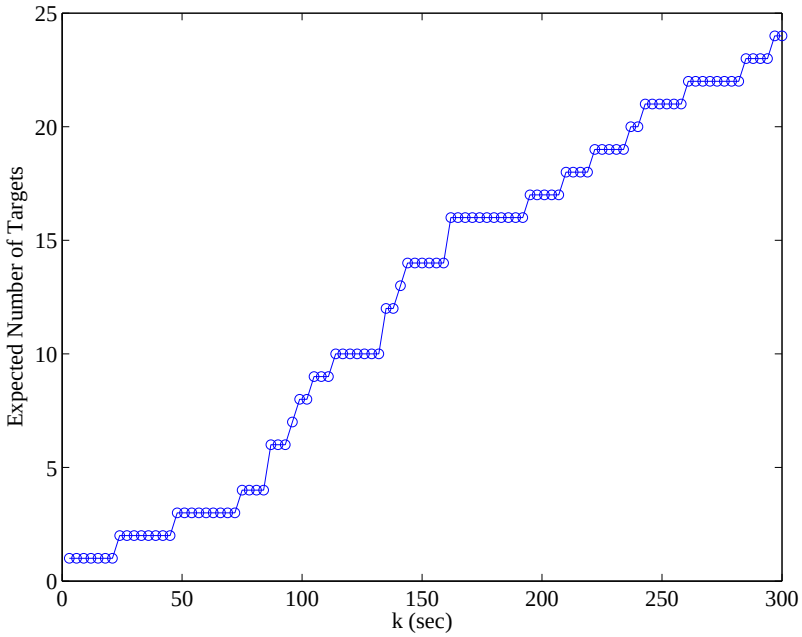



**Figure 43:** Expected number of targets at each time step in Experiment 2 with multipath effects, where the PHD filter uses the smallest multipath power ratio to compute the $p_D$ and SNR.

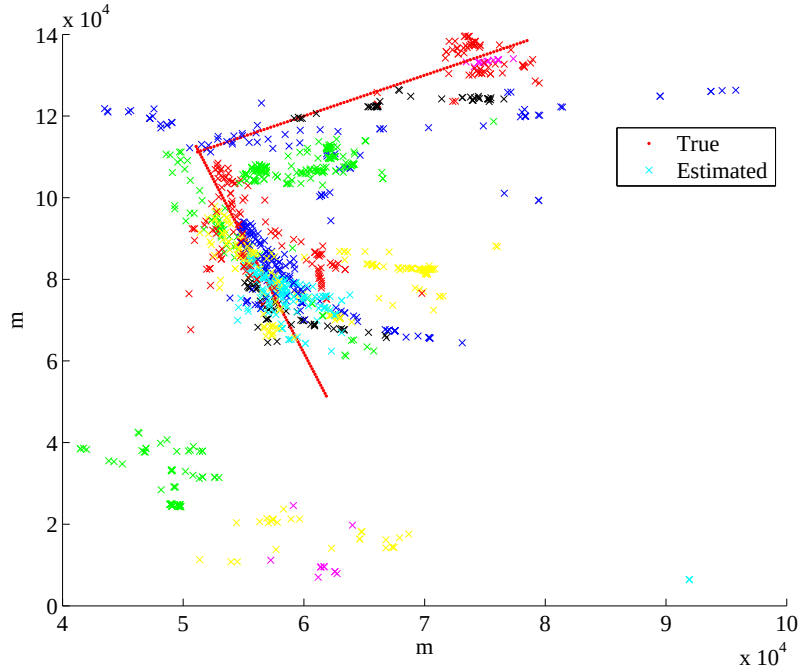



**Figure 44:** Actual vs. estimated target locations over the first 300 iterations in Experiment 2 with multipath effects, , where the PHD filter uses the smallest multipath power ratio to compute the $p_D$ and SNR.

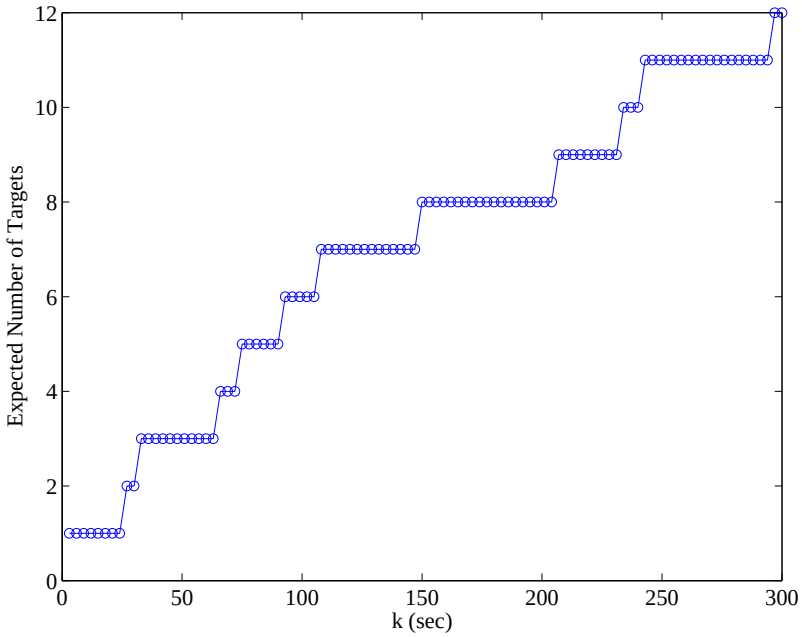



**Figure 45:** Expected number of targets at each time step in Experiment 2 with multipath effects and false alarm suppression, where the PHD filter uses the smallest multipath power ratio to compute the $p_D$ and SNR.

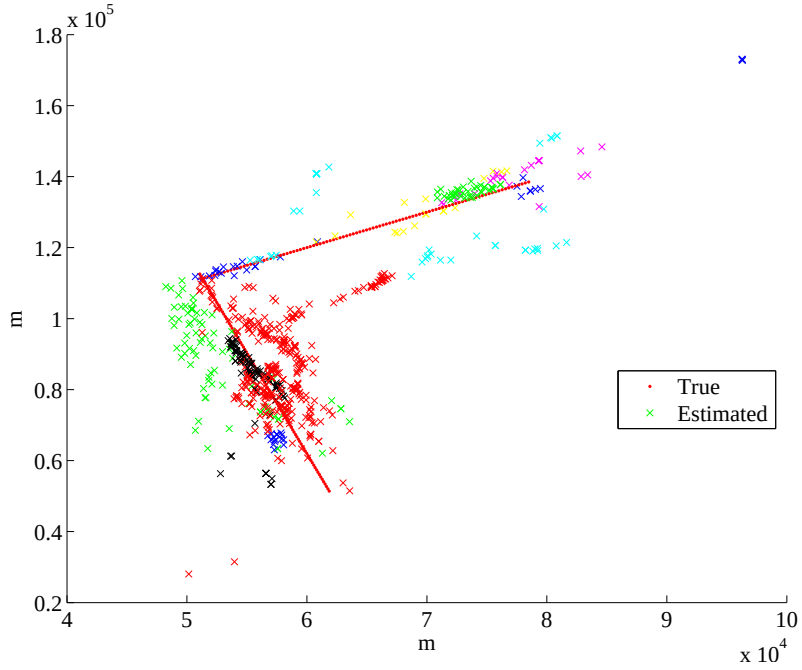



**Figure 46:** Actual vs. estimated target locations over the first 300 iterations in Experiment 2 with multipath effects and false alarm suppression, where the PHD filter uses the smallest multipath power ratio to compute the $p_D$ and SNR.

**Experiment 3** The largest possible $p_D$ values were now used in the Data Update step of the PHD filter by assuming the best-case multipath power ratio, i.e., $\max(F_R^2 F_T^2)$, for the entire FoV. The result, as seen in Figures 47-49, was that the number of targets was rarely overestimated, but instead the PHD filter had a hard time detecting both targets. This was expected, given the results of Experiment 2, since in this case, the PHD filter assumed an SNR that was too high, and the range variance used to compute the likelihood was too small. Thus, the filter most likely missed valid observation intersections.

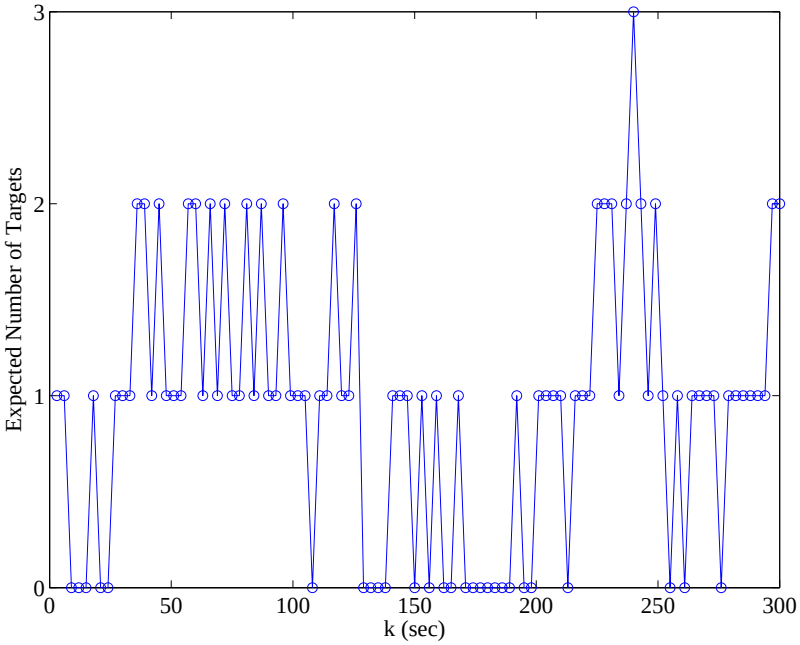



**Figure 47:** Expected number of targets at each time step in Experiment 3 with multipath effects, where the PHD filter uses the largest multipath power ratio to compute the $p_D$ and SNR.

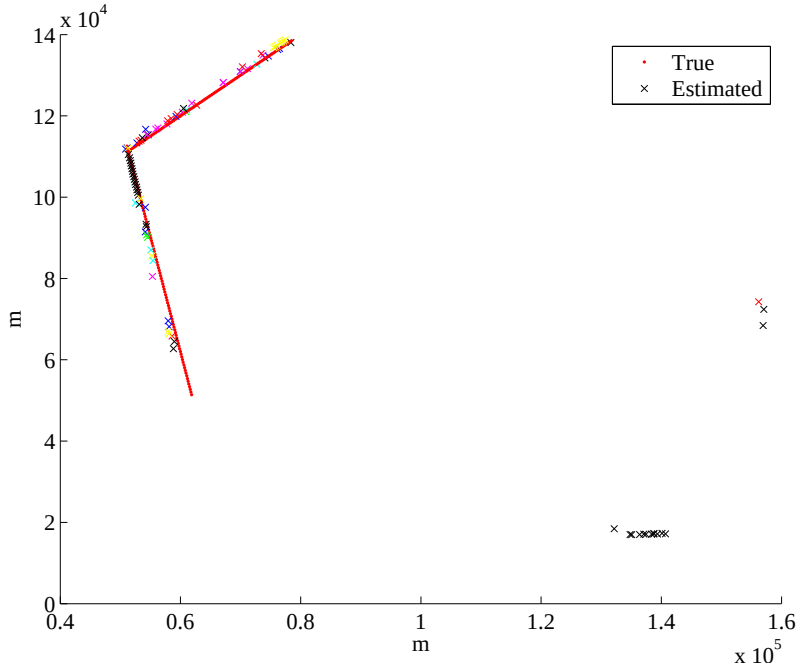
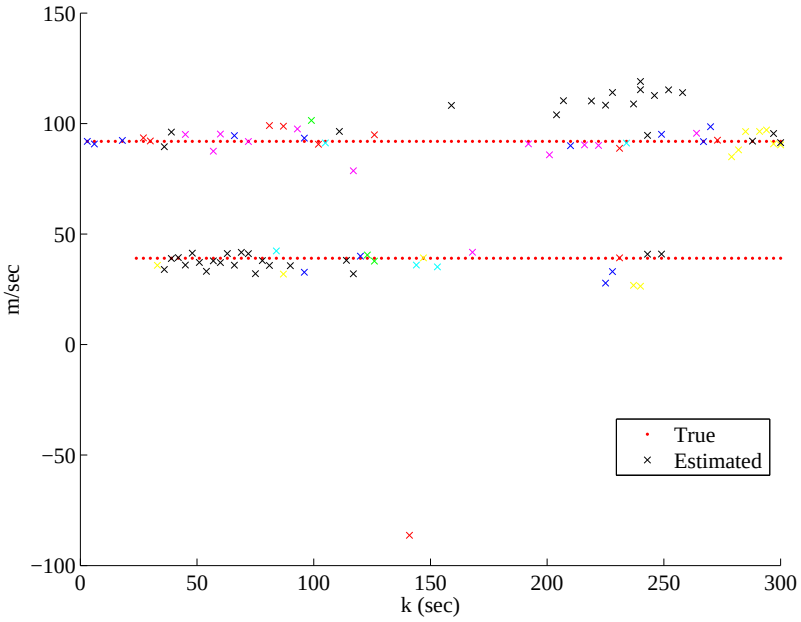
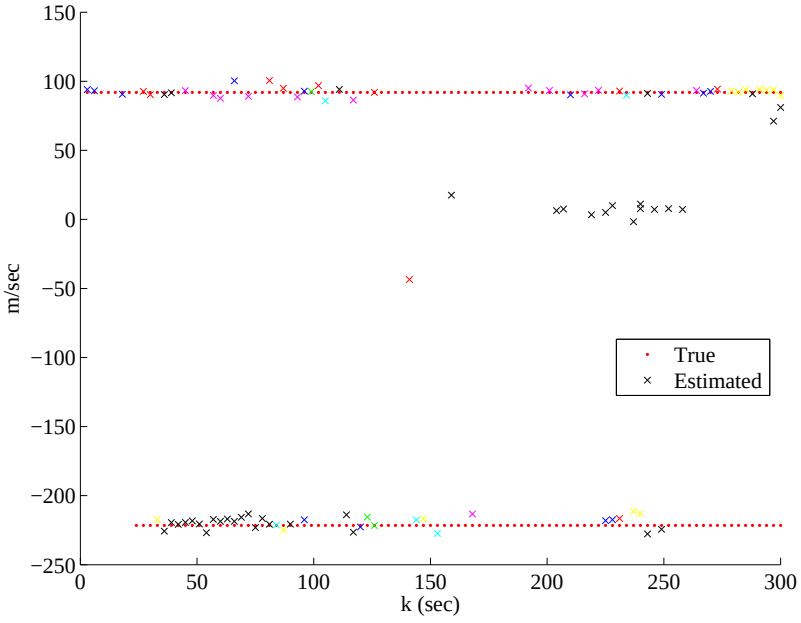



**Figure 48:** Actual vs. estimated target locations over the first 300 iterations in Experiment 3 with multipath effects, where the PHD filter uses the largest multipath power ratio to compute the $p_D$ and SNR.




(a) $\dot{x}$




(b) $\dot{y}$

**Figure 49:** Actual vs. estimated target velocities over the first 300 iterations in Experiment 3 with multipath effects, where the PHD filter uses the largest multipath power ratio to compute the $p_D$ and SNR.

**Experiment 4** In this simulation, it was assumed that the PHD filter did not know the true $p_D$ and SNR values to use in the Data Update step. However, the offline range-variance based grid used to find the ellipse intersections was recomputed assuming that it knew the true target altitude. This recomputed grid, which uses the correct multipath-dependent SNR, is shown in Figure 50. The result of the simulation is depicted in Figures 51-53. Both targets are lost and a ghost target is tracked, where the $p_D$ drops significantly in one or more of the transmitters used. The PHD filter again occasionally overestimates the number of targets present. This poor performance is expected, since even though the birth particles are being placed according to the multipath truth, the sensor likelihoods used in the PHD filter to compute the particle weights do not correctly model the multipath.

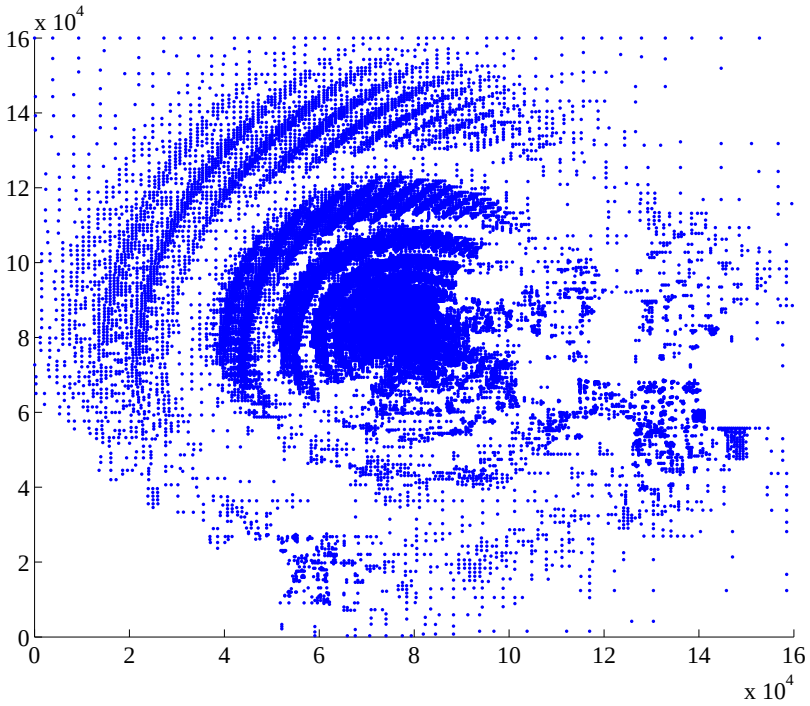



**Figure 50:** Gridpoint locations of the bistatic range variance-based grid computed offline using the multipath truth for the SNR to calculate $\sigma_R$. These gridpoints indicate the centers of the gridspaces used to locate ellipse intersections and distribute birth particles. The gridpspaces are $4\sigma_R \times 4\sigma_R$ and at least 10 meters wide, and they generally overlap each other by $2\sigma_R$. The gridpoint centers are no closer than $1.7\sigma_R$.

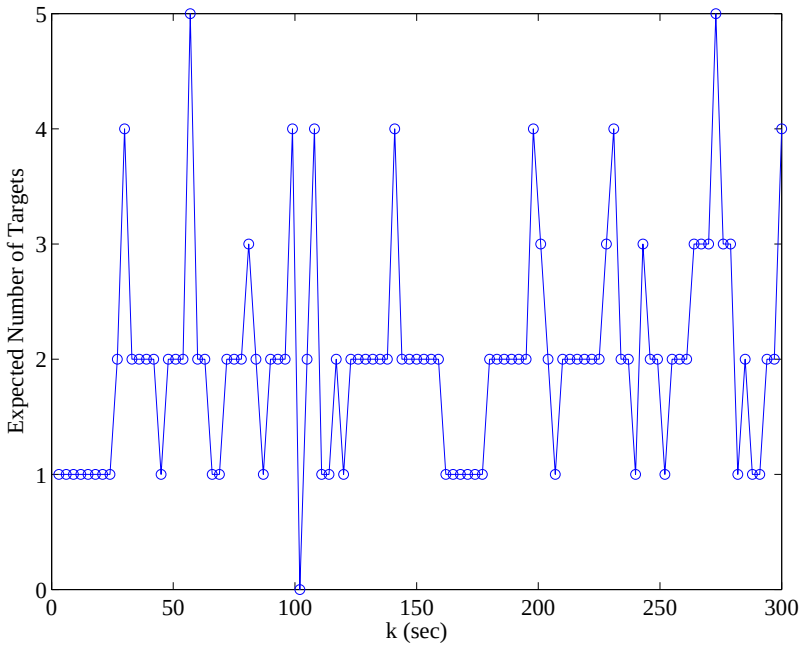



**Figure 51:** Expected number of targets at each time step in Experiment 4 with multipath effects, where the precomputed range-variance based grid knows the multipath truth.

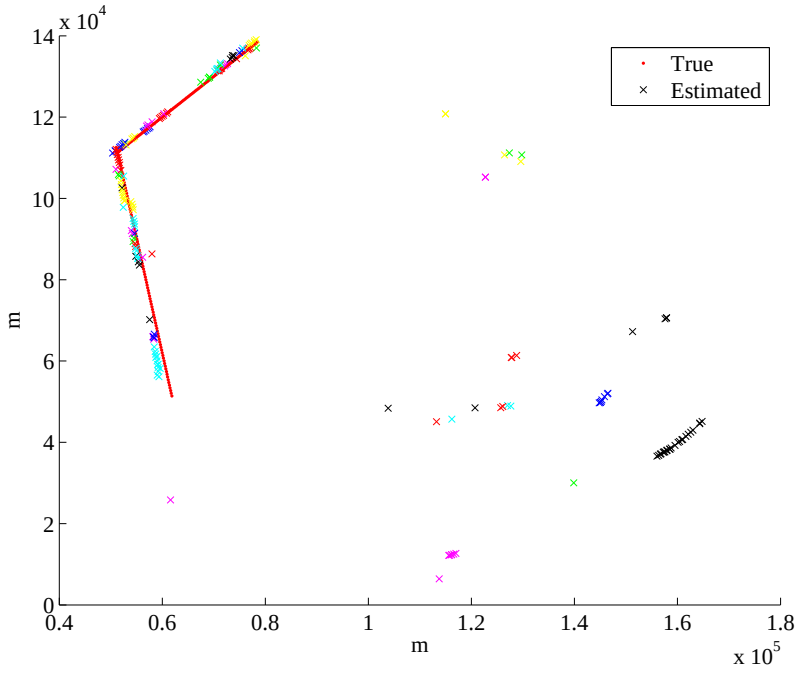
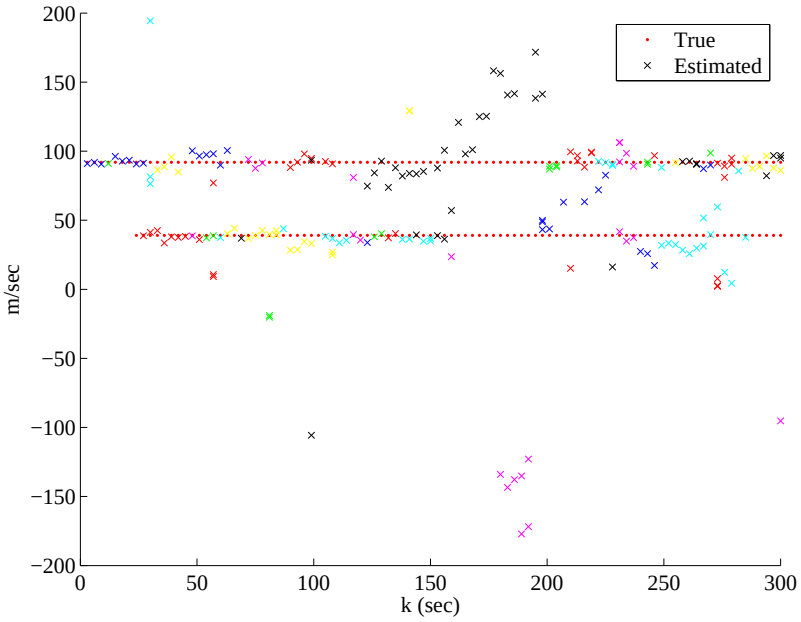
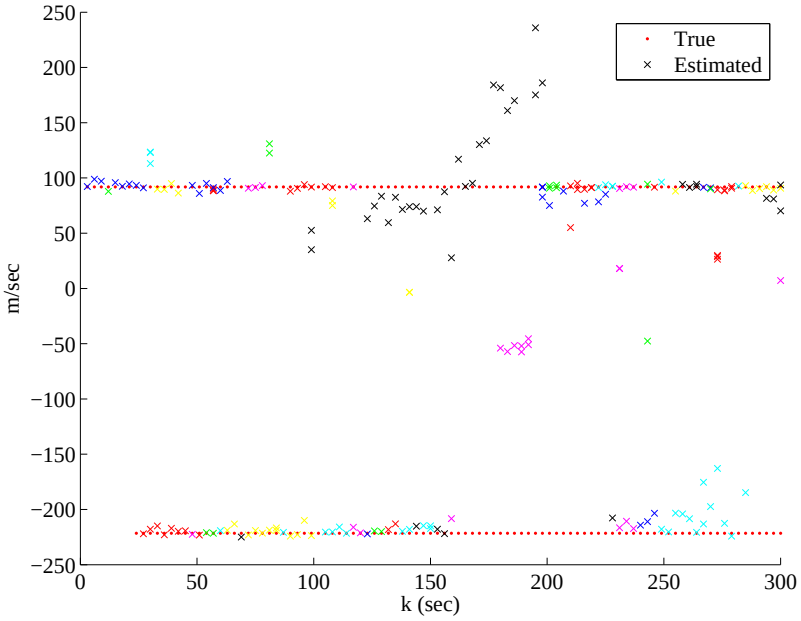



**Figure 52:** Actual vs. estimated target locations over the first 300 iterations in Experiment 4 with multipath effects, where the precomputed range-variance based grid knows the multipath truth.




(a) $\dot{x}$




(b) $\dot{y}$

**Figure 53:** Actual vs. estimated target velocities over the first 300 iterations in Experiment 4 with multipath effects, where the precomputed range-variance based grid knows the multipath truth.

**Experiment 5** Both the ellipse-intersection finding grid and the Data Update step were assumed to know the correct $p_D$ and SNR to use in this simulation. The result, as seen in Figures 54-56, was that Target 1 was detected most of the time, Target 2 was undetected while in the region of low SNR, and the overestimation of the number of targets was as severe as in experiment 1 when just the PHD filter knew the multipath truth.

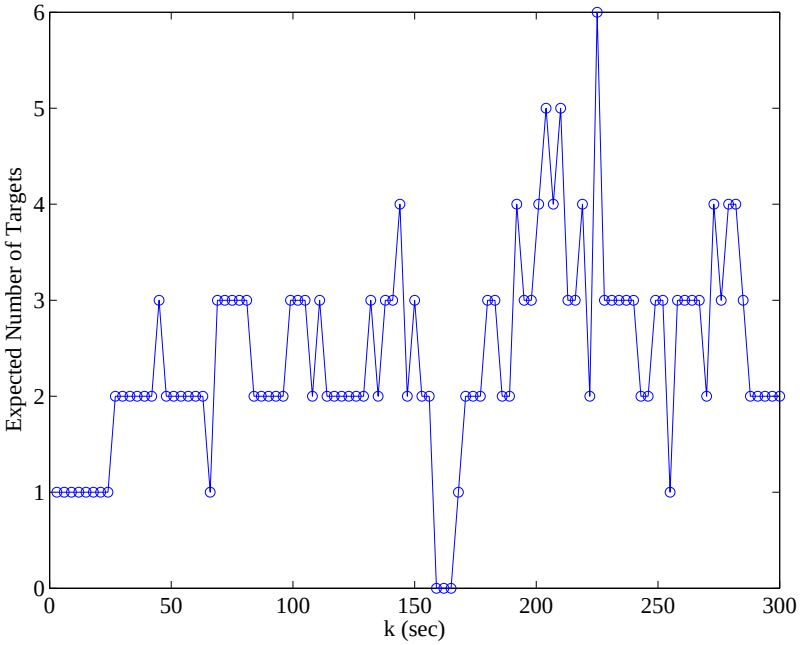



**Figure 54:** Expected number of targets at each time step in Experiment 5 with multipath effects, where both the precomputed range-variance based grid and the PHD filter know the multipath truth.

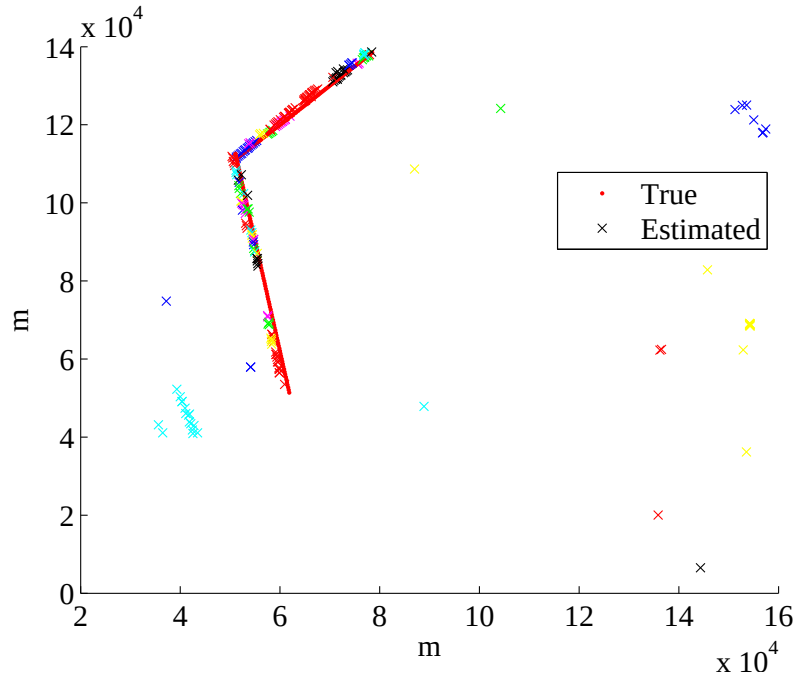
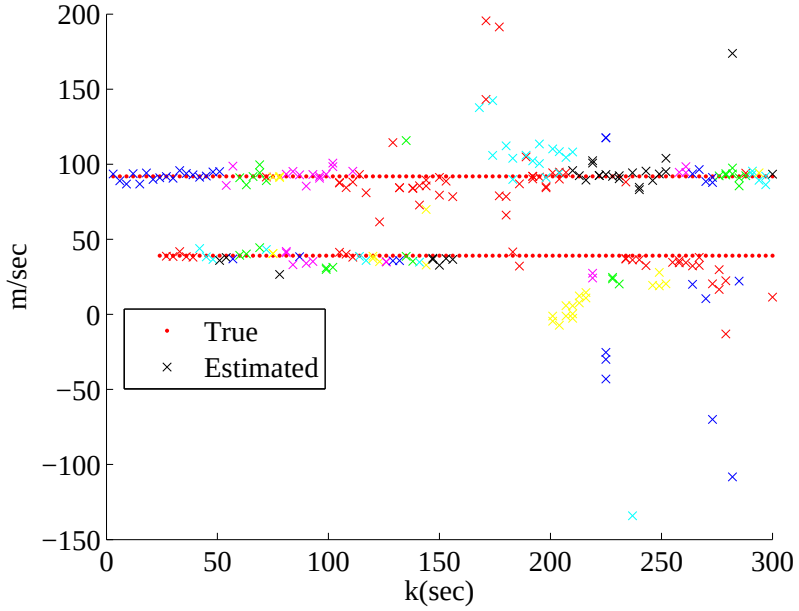
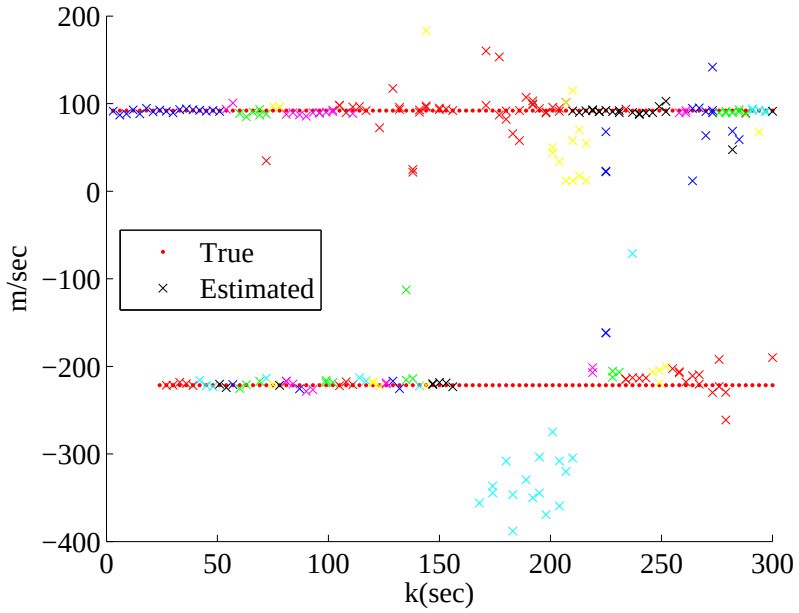



**Figure 55:** Actual vs. estimated target locations over the first 300 iterations in Experiment 5 with multipath effects, where both the precomputed range-variance based grid and the PHD filter know the multipath truth.




(a) $\dot{x}$




(b) $\dot{y}$

**Figure 56:** Actual vs. estimated target velocities over the first 300 iterations in Experiment 5 with multipath effects, where both the precomputed range-variance based grid and the PHD filter know the multipath truth.

**Experiment 6** The original multipath simulation, where neither the PHD filter, nor the range-variance based grid, know the true SNR or $p_D$ caused by the multipath, was re-run. The false alarm parameters used were kept the same, as well. However, false alarms were prevented from occurring during the simulation run. The results are shown in Figures 57-59. Just as in the original multipath simulation (see Figures 32-36), the PHD filter fails to detect the two targets while they travel in areas of low SNR. However, the current results show almost no overestimation of the expected number of targets, and considerably fewer ghost targets are detected than in the original multipath simulation. However, since the false alarm parameters specified do indicate the presence of false alarms, the PHD filter expects false alarms, and we expect it to underestimate slightly the number of targets.

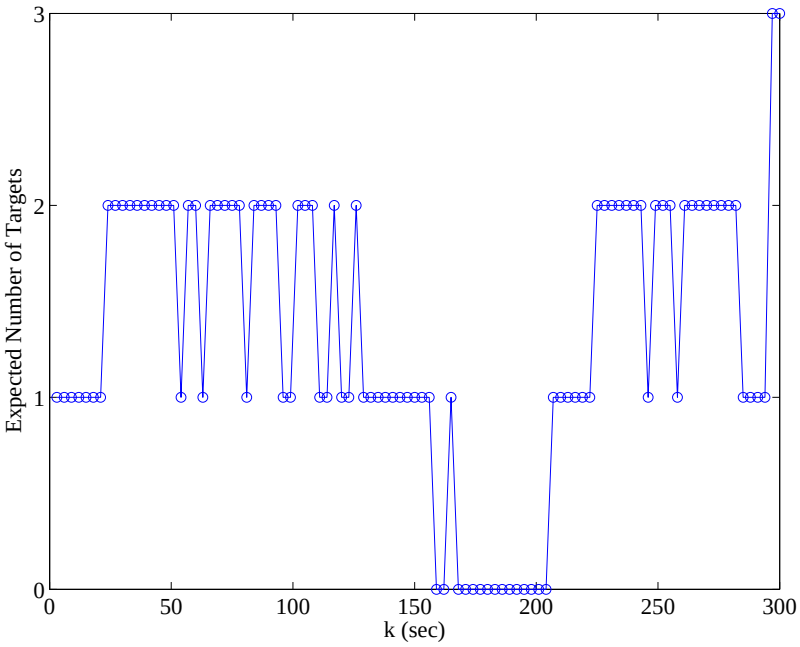



**Figure 57:** Expected number of targets at each time step in Experiment 6 with multipath effects and false alarm suppression.

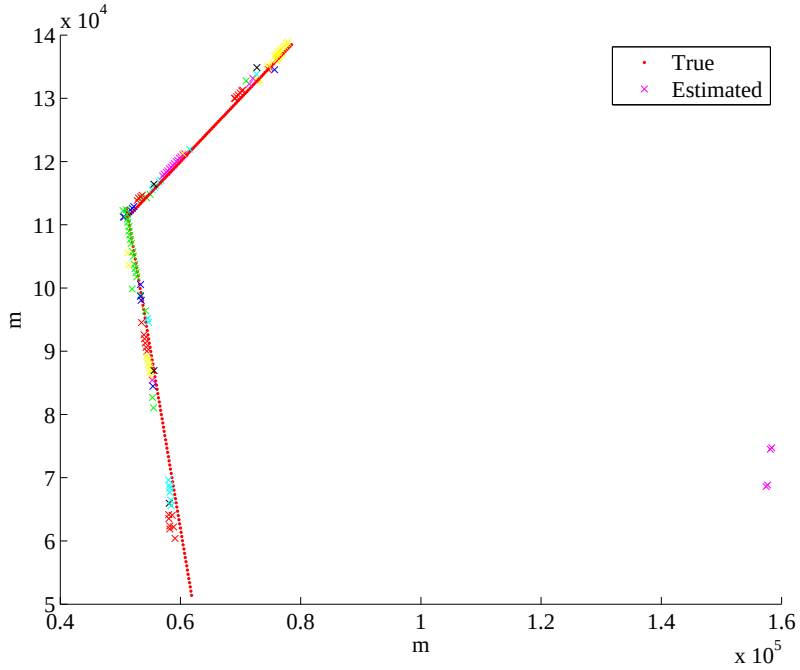
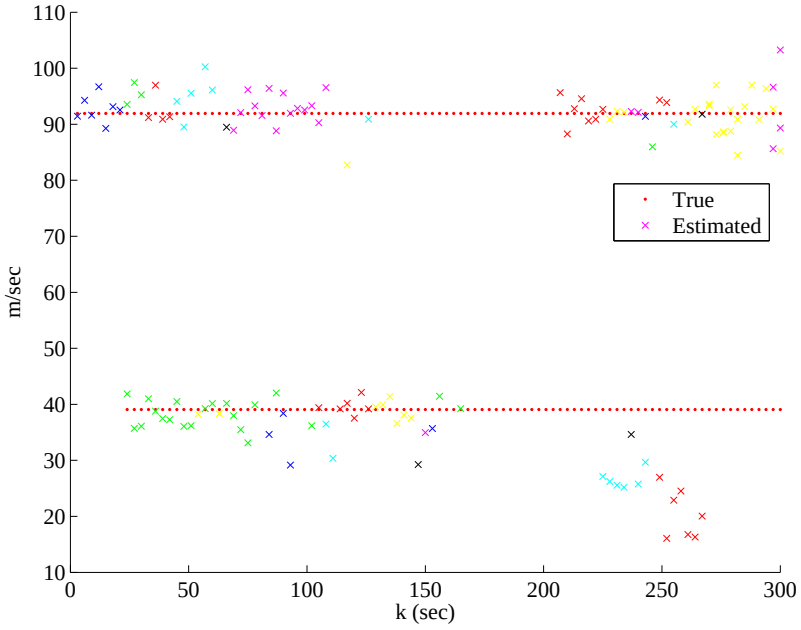
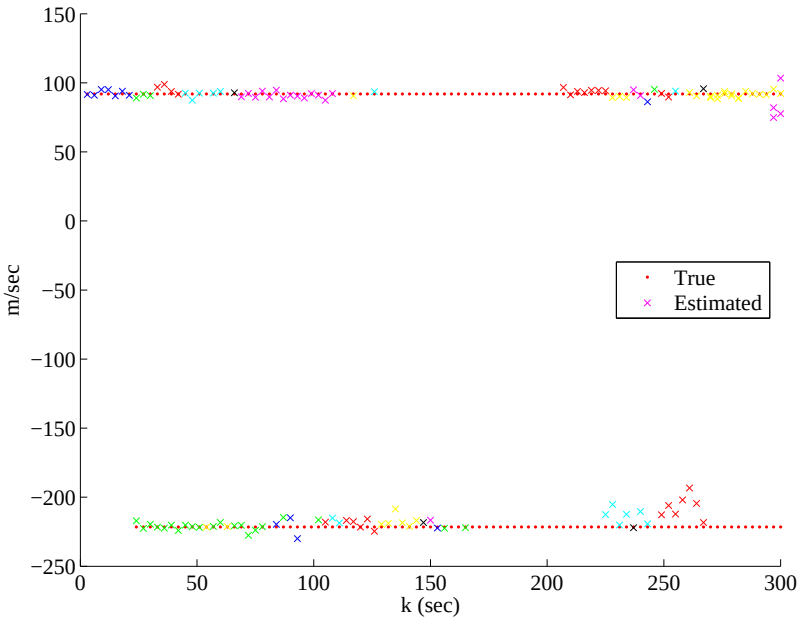



**Figure 58:** Actual vs. estimated target locations over the first 300 iterations in Experiment 6 with multipath effects and false alarm suppression.




(a) $\dot{x}$




(b) $\dot{y}$

**Figure 59:** Actual vs. estimated target velocities over the first 300 iterations in Experiment 6 with multipath effects and false alarm suppression.

**Experiment 7** The previous experiment with suppressed false alarms was repeated, but the PHD filter and range-varianced based grid were given knowledge of the true $p_D$ and SNR caused by the multipath, as in Experiment 5. The results are shown in Figures 60-62. As in Experiment 5, Target 2 was not detected in the region of low SNR. Target 1 was detected almost always, which is slightly more than in Experiment 5. The main differences between this simulation and Experiment 5 are the almost complete absence of ghost targets and the much better performance in target number estimation.

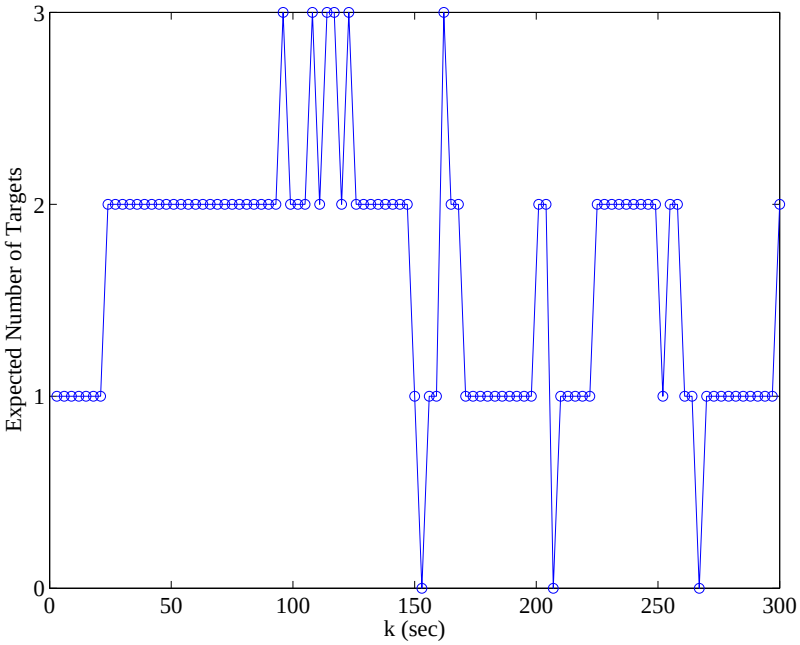



**Figure 60:** Expected number of targets at each time step in Experiment 7 with multipath effects and false alarm suppression, and where both the precomputed range-variance based grid and the PHD filter know the multipath truth.

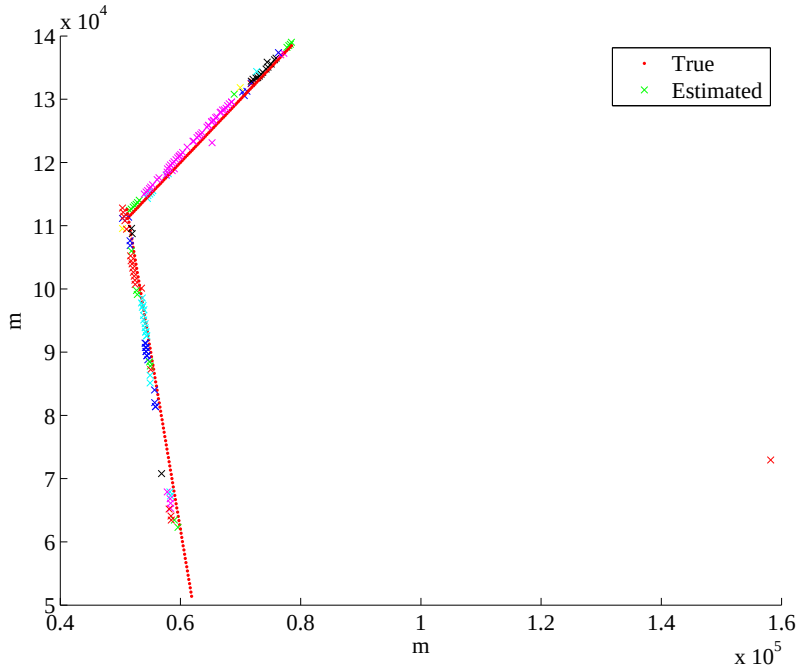
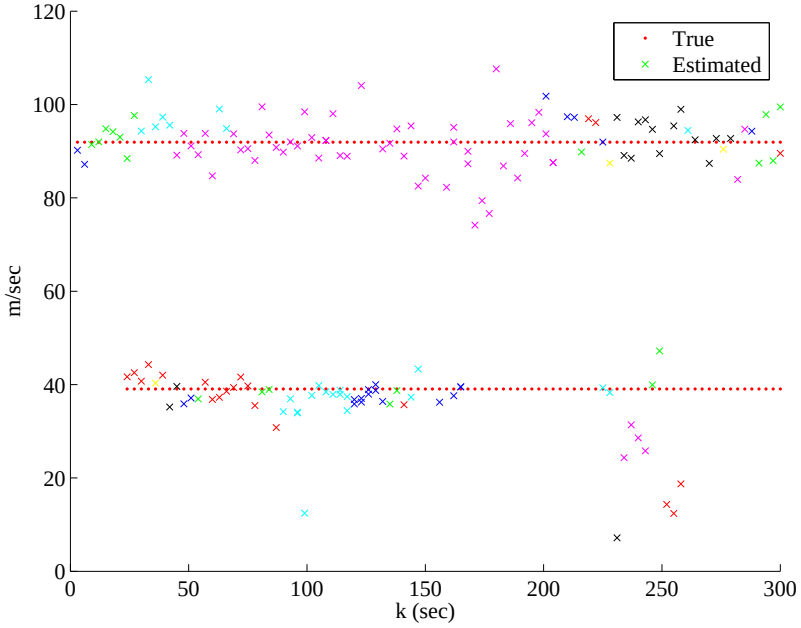
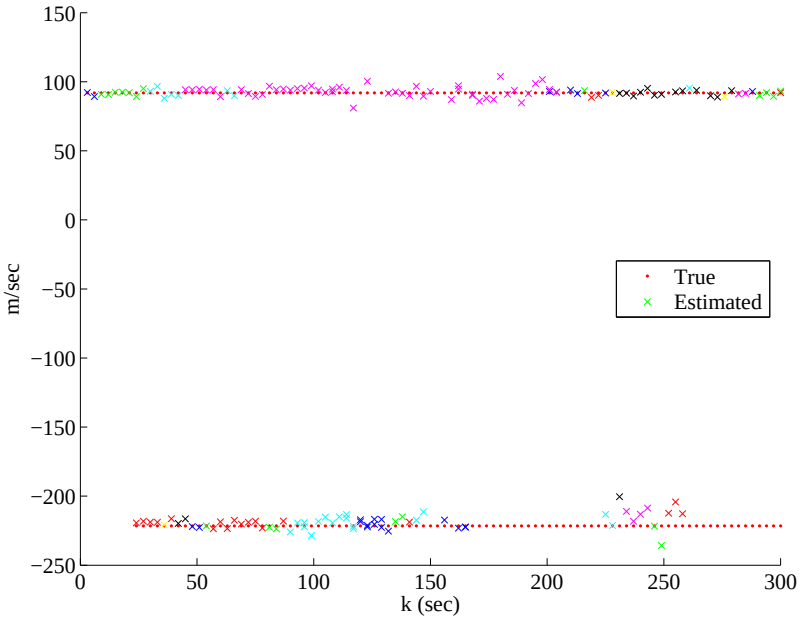



**Figure 61:** Actual vs. estimated target locations over the first 300 iterations in Experiment 7 with multipath effects and false alarm suppression, and where both the precomputed range-variance based grid and the PHD filter know the multipath truth.




(a) $\dot{x}$




(b) $\dot{y}$

**Figure 62:** Actual vs. estimated target velocities over the first 300 iterations in Experiment 7 with multipath effects and false alarm suppression, and where both the precomputed range-variance based grid and the PHD filter know the multipath truth.

### 6.4.3 Conclusions

The simulation experiments in the previous section give us some insight into the robustness of the PHD filter. It is seen that when the actual probabilities of detection and signal to noise ratios differ from those assumed by the PHD filter, then target tracking performance suffers. We find that it is not suffering via birth particle placement issues, since adding multipath truth information into the range-variance based grid did not affect the simulation results. It is the incorrectness in the sensor likelihood functions (see Section 4.4.3), caused by the discrepancy between the true $p_D$ and SNR and the $p_D$ and SNR assumed by the PHD filter, that affects the tracking performance. This is evident multiple times in the experiments of the previous section, as discussed below.

In the case where the PHD filter overestimates the SNR, as in Experiment 3, the sensor likelihood function is too tight, and the PHD filter has a difficult time detecting the targets. In the case where the PHD underestimates the SNR, as in Experiment 2, the sensor likelihood function is too loose, and the PHD filter overestimates the number of targets present, even in the absence of false alarms. Another reason for the overestimation is that the PHD does not expect to see the targets given the underestimated SNR. Yet, because the SNR is really high, it still receives target observations and thus overestimates the number of targets present. When the true values of $p_D$ and SNR were given to the PHD filter, as in Experiments 1 and 5, the tracking performance improved. Only one of the targets remained undetected when traveling through an area of low SNR.

Experiments 6 and 7 attempted to address the issue of the target number overestimation by the PHD filter, which is present in these multipath simulations even when the PHD filter knows the true $p_D$ and SNR values. When false alarms were suppressed in the simulation, the overestimation problem, as well as the detection of ghost targets, almost vanished. This leads us to believe that the overestimation problem is different from that observed due to the $(1 - p_D)$ term in the Data Update equation of the PHD filter when spontaneously placing birth particles in areas of low $p_D$ (see Sections 4.6.1, 5.2.7, and 6.3). The current overestimation problem, when the true $p_D$ and SNR are known, appears to be due to an interplay between false alarms and the ambiguity in triangulating targets for which not

all sensors have high $p_D$. Thus, in the presence of false alarms, the PHD filter is highly sensitive to the probabilities of detection and signal-to-noise ratios present in the tracking scenario.

## *6.5 Direction of Arrival Observations*

### 6.5.1 Introduction

In the hope of improving the performance of the multitarget tracker, directon of arrival (DOA) measurements were incorporated into the simulation as another form of observation data available to the PHD filter at each time step. DOA measurements specify the angle at which the incoming reflected radio wave arrives at the receiver. To obtain the DOA, an antenna array is necessary to measure the difference in phase of the incoming wave at the different elements in the array. The configuration specified in [28] is used and consists of two vertically-polarized, half-wave dipole antennas separated a half-wavelength's distance apart. The phase difference, $\Phi$, in the two dipoles is used to find the angle of arrival:

$$\theta_{doa} = \sin^{-1}\left(\frac{\lambda\Phi}{2\pi d}\right), \tag{157}$$

where $\lambda$ is the wavelength, $d$ is the antenna spacing, and where it is assumed that $-\pi < \Phi < \pi$ to avoid directional amibiguities [26]. With an antenna separation of $d = \frac{\lambda}{2}$, the range of observable DOA values varies over $-90^\circ < \theta_{doa} < 90^\circ$, and there is an ambiguity of $180^\circ$ present in the DOA measurements. That is, one cannot tell from a DOA observation whether a target is in front of the antenna or behind it.

Assuming that the noise on each antenna has a Rayleigh-distributed amplitude and uniformly-distributed noise, Howland calculates the effect of the SNR on the DOA measurements [26]. He finds that for a target with a true DOA of $\theta_{true}$ and a given SNR at the antenna inputs, the probability density function (PDF) of the observed DOA is[34]

$$P_{\theta_{doa}}(\theta_{doa}) = \frac{d\sqrt{2\pi\mathrm{SNR}}}{\lambda}\cos(\theta_{doa})\cdot e^{-\frac{\mathrm{SNR}}{2}\left(\frac{2\pi d}{\lambda}\right)^2(\sin\theta_{doa}-\sin\theta_{true})^2}. \tag{158}$$

[3]This is a correction to (1.17) in [26]. There is a typo in (1.12) in [26] (*viz.*, the denominator in the exponent should read '$4\sigma^2$', not '$2\sigma^2$') that propagates to (1.17), which should read '$\frac{S}{2}$', not '$S$', in the exponent.

[4]Technically, we should write $P_{\theta_{doa}}(\theta_{doa})$ as $P_{\theta_{doa}}(\theta_{doa}|\theta_{true})$ and its cumulative distribution function, $F_{\theta_{doa}}(\theta_{obs})$, as $F_{\theta_{doa}}(\theta_{obs}|\theta_{true})$.

The value of $P_{\theta_{doa}}(\theta_{doa})$ for various values of $\theta_{true}$, for an SNR of 18dB and $d = \frac{\lambda}{2}$, is illustrated in Figure 63. Note that these PDFs are not symmetric and that the variance of the DOA observation increases as the true target DOA gets farther away from boresight.

### 6.5.2 DOA Likelihood, $f_{doa}(\mathbf{z}|\boldsymbol{\xi})$

To incorporate the DOA observations into the PCL simulation, we need to derive the single-target sensor likelihood function $f_{doa}(\mathbf{z}|\boldsymbol{\xi})$ for use in the Data Update stage of the PHD filter (see Section 4.4.3), where the observation $\mathbf{z}$ is the DOA measurement $\theta_{doa}$, and the particle state $\boldsymbol{\xi}$ is used to calculate the SNR and $\theta_{true}$. The wavelength used for the antenna separation is that of the Lopik transmitter (see Table 11), i.e., $d = \frac{\lambda_{Lopik}}{2}$.

We note that the PDF given by (158) is a Gaussian in terms of $\sin\theta_{doa}$, and so we can construct the likelihood function by using its associated cumulative distribution function (CDF). The CDF is found by taking the integral of (158):

$$F_{\theta_{doa}}(\theta_{obs}) = \int_{-90^\circ}^{\theta_{obs}} P_{\theta_{doa}}(\theta_{doa}) d\theta_{doa}, \tag{159}$$

where $-90^\circ < \theta_{obs} < 90^\circ$, so that $\sin(\theta_{obs})$ is unique over this integral,

$$= \int_{-90^\circ}^{\theta_{obs}} \frac{d\sqrt{2\pi\mathrm{SNR}}}{\lambda} \cos(\theta_{doa}) \cdot e^{-\frac{\mathrm{SNR}}{2}\left(\frac{2\pi d}{\lambda}\right)^2 (\sin\theta_{doa} - \sin\theta_{true})^2} d\theta_{doa}, \tag{160}$$

$$= \int_{\sin(-90^\circ)}^{\sin(\theta_{obs})} \frac{d\sqrt{2\pi\mathrm{SNR}}}{\lambda} \cdot e^{-\frac{\mathrm{SNR}}{2}\left(\frac{2\pi d}{\lambda}\right)^2 (\nu - \sin\theta_{true})^2} d\nu, \tag{161}$$

where $\nu = \sin(\theta_{doa})$ and $d\nu = \cos(\theta_{doa}) d\theta_{doa}$, (162)

$$= \int_{-1}^{\nu_{obs}} \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(\nu - \sin\theta_{true})^2}{2\sigma^2}} d\nu, \tag{163}$$

where $\nu_{obs} = \sin(\theta_{obs})$, and $\sigma^2 = \frac{1}{\mathrm{SNR}} \left(\frac{\lambda}{2\pi d}\right)^2$. (164)

The term in the integral of (163) is a Gaussian with a mean of $\sin(\theta_{true})$ and variance $\sigma^2$ given in (164). Assuming $-90^\circ < \theta_{true} < 90^\circ$, by splitting the integral and using the Matlab error function,

$$\mathrm{erf}(\chi) = \frac{2}{\pi} \int_0^{\chi} e^{-t^2} dt, \tag{165}$$

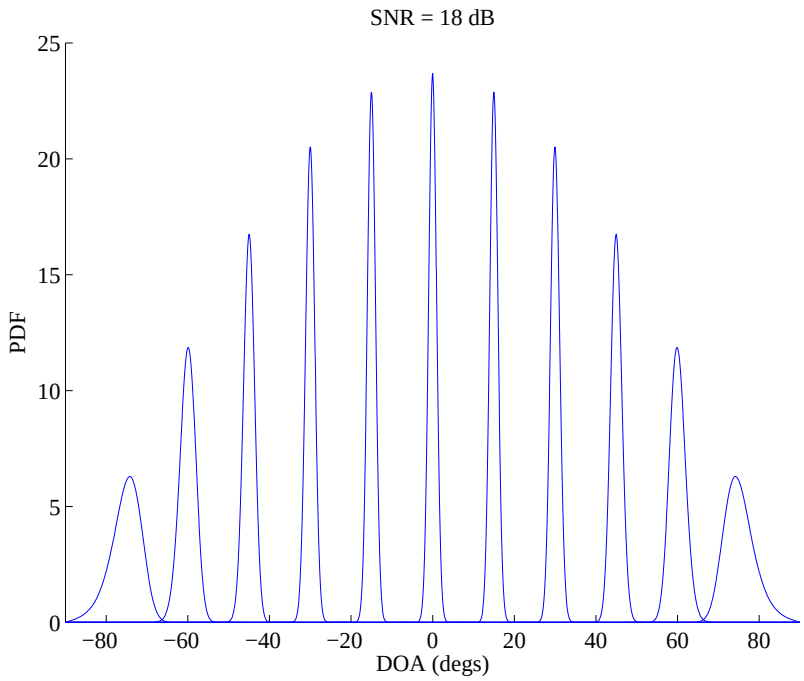



**Figure 63:** The PDFs, $P_{\theta_{doa}}(\theta_{doa})$, of the direction of arrival measurement is shown for various values of $\theta_{true}$ in the case of an 18dB SNR and antenna separation of half a wavelength (i.e., $d = \frac{\lambda}{2}$). Each curve in the figure represents the PDF of $\theta_{doa}$ for a different value of $\theta_{true}$. Starting with the left-most curve, the values used for $\theta_{true}$ to generate each curve are: $-75°, -60°, -45°, -30°, -15°, 0°, 15°, 30°, 45°, 60°,$ and $75°$. This figure is adapted from [26].

the CDF can be written as:

$$F_{\theta_{doa}}(\theta_{obs}) = \frac{1}{\sqrt{2\pi\sigma^2}} \left[ \int_{\sin(\theta_{true})}^{\nu_{obs}} e^{-\frac{(\nu-\sin\theta_{true})^2}{2\sigma^2}} d\nu + \int_{-1}^{\sin(\theta_{true})} e^{-\frac{(\nu-\sin\theta_{true})^2}{2\sigma^2}} d\nu \right] \quad (166)$$

$$= \frac{\text{erf}}{2}\left(\frac{\nu_{obs} - \sin(\theta_{true})}{\sigma\sqrt{2}}\right) + \frac{\text{erf}}{2}\left(\frac{\sin(\theta_{true}) - (-1)}{\sigma\sqrt{2}}\right). \quad (167)$$

In the case of high SNR, the variance of the Gaussian is small, and

$$\tilde{F}_{\theta_{doa}} = \int_{\sin(-90^\circ)}^{\sin(90^\circ)} \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(\nu-\sin\theta_{true})^2}{2\sigma^2}} d\nu \approx 1. \quad (168)$$

However, when SNR is low, the variance may be large enough such that $P_{\theta_{doa}}(\theta_{doa})$ is not a valid PDF over the interval $-90^\circ < \theta_{doa} < 90^\circ$. In this case, the PDF is not Gaussian with respect to $\nu$, and $\tilde{F}_{\theta_{doa}} \neq 1$. Nevertheless, instead of simply assuming a uniform random distribution for $\theta_{obs}$ over $[-90^\circ, 90^\circ]$, we still use the PDF, so that some of the structure of the distribution of $\theta_{obs}$ around $\theta_{true}$ is maintained. To achieve this, we normalize $F_{\theta_{doa}}(\theta_{obs})$ by $\tilde{F}_{\theta_{doa}}$, such that

$$\overline{F}_{\theta_{doa}}(\theta_{obs}) = \frac{F_{\theta_{doa}}(\theta_{obs})}{\tilde{F}_{\theta_{doa}}}$$

$$= \frac{1}{\tilde{F}_{\theta_{doa}}} \left[ \frac{\text{erf}}{2}\left(\frac{\nu_{obs} - \sin(\theta_{true})}{\sigma\sqrt{2}}\right) + \frac{\text{erf}}{2}\left(\frac{\sin(\theta_{true}) + 1}{\sigma\sqrt{2}}\right) \right] \quad (169)$$

integrates to unity over $[\sin(-90^\circ), \sin(90^\circ)]$, and where, using (165) and (168), the normalizing constant can be rewritten as

$$\tilde{F}_{\theta_{doa}} = \frac{\text{erf}}{2}\left(\frac{1 - \sin(\theta_{true})}{\sigma\sqrt{2}}\right) + \frac{\text{erf}}{2}\left(\frac{\sin(\theta_{true}) + 1}{\sigma\sqrt{2}}\right). \quad (170)$$

To calculate the single-target likelihood function for the DOA measurement, we use the property of the CDF [63], whereby the probability that the observed DOA measurement is equal to the target's true DOA is given by:

$$f_{doa}(\mathbf{z}|\boldsymbol{\xi}) = \Pr(\theta_{obs} - \epsilon < \theta_{true} \leq \theta_{obs}) \quad (171)$$

$$= \overline{F}_{\theta_{doa}}(\theta_{obs}) - \overline{F}_{\theta_{doa}}(\theta_{obs} - \epsilon), \quad (172)$$

where $\epsilon$ is taken to be $0.1^\circ$ in our simulations[5]. The particle state $\boldsymbol{\xi}$ is used to calculate the SNR, via (122), as well as the $\theta_{true}$ relative to the receiver boresight, $\theta_{boresight}$, in the

[5]In the case where $\theta_{obs} - \epsilon < -90^\circ$, the following is used: $f_{doa}(\mathbf{z}|\boldsymbol{\xi}) = \overline{F}_{\theta_{doa}}(\theta_{obs} + \epsilon) - \overline{F}_{\theta_{doa}}(\theta_{obs})$.

following manner:

$$\theta_{target} = -\tan^{-1}\left(\frac{y - y_r}{x - x_r}\right) + 90^\circ, \tag{173}$$

and

$$\theta_{true} = \theta_{target} - \theta_{boresight}, \tag{174}$$

where $(x, y)$ is the location specified by the particle, and $(x_r, y_r)$ is the location of the receiver. Recalling that we have a 180° ambiguity in the DOA measurements, $\theta_{true}$ is adjusted by 180° or 360°, if necessary, so that $-90^\circ < \theta_{truth} < 90^\circ$.

### 6.5.3 Generating DOA Observations

Solving for $\nu_{obs}$ in the normalized cumulative probability distribution for DOA measurements given in (169), we obtain:

$$\nu_{obs} = \sin(\theta_{true}) + \sigma\sqrt{2} \cdot \text{erfinv}\left[2\tilde{F}_{\theta_{doa}}\overline{F}_{\theta_{doa}}(\theta_{obs}) - \text{erf}\left(\frac{\sin(\theta_{true}) + 1}{\sigma\sqrt{2}}\right)\right], \tag{175}$$

where erfinv$(\cdot)$ is the inverse function of (165). To generate the observed DOA for a $\theta_{true}$ in the simulation, a random number is chosen from a uniform distribution over $(0, 1)$ and used as the value of the normalized CDF, $\overline{F}_{\theta_{doa}}(\theta_{obs})$, in (175). The observed DOA obtained is then

$$\theta_{obs} = \sin^{-1}(\nu_{obs}). \tag{176}$$

Again, recalling the 180° ambiguity in the DOA measurements, a target's $\theta_{true}$ may need to be adjusted by 180° or 360°, so that $-90^\circ < \theta_{truth} < 90^\circ$.

### 6.5.4 Simulation Results

DOA observations and the DOA sensor likelihood were incorporated into the initial two-target, multipath-included simulation of Section 6.4.2. The result of tracking the two targets using range, Doppler, and DOA observations is shown in Figures 64-67. Comparing these figures to those of Figures 32-36, in which only range and Doppler observations were used in the multipath simulation, one sees that there were fewer target detections when the DOA measurements were added to the range and Doppler observations. Although the range, Doppler, and DOA tracker failed to detect the true targets as often as the tracker

without DOA, on a positive note, the DOA-inclusive tracker did not produce a single ghost target detection. The same result is obtained when multipath effects are removed from the range-Doppler and range-Doppler-DOA simulations.

### 6.5.5 Conclusions

The results of adding DOA to the range and Doppler tracking present a seeming curiosity. We would expect to obtain better performance by adding more observation information about the targets into the tracker. Adding DOA measurements did improve the tracker's defense against detecting ghost targets; however, it worsened the tracker's ability to detect the true targets. The reason is most likely a particle sampling issue. Increasing the number of particles used in the DOA simulation by a factor of ten improved the target detection, as seen in Figure 68, but also introduced ghost target detections, as evident in Figure 69. In Chapter 4, to improve the performance of the tracker, we needed to propose smarter birth particle placement, instead of throwing more particles into the filter. Finding the best states at which to place the particles involved taking the range and Doppler observations into account, namely, by finding their intersections. It appears that a similar approach towards the DOA is necessary here. That is, a technique for placing particles that incorporates the DOA observations by exploiting the DOA information appears necessary to improve tracking performance. This remains an avenue for future work.

## *6.6 Accelerating Target Example*

In all the simulations run, so far, the targets have been traveling with a constant velocity. To test the robustness of the current implementations, we add acceleration to Target 1. In the following simulations, Target 1 starts with a velocity of 130 m/sec (468 km/h) with a Northeast heading of 45°. At every iteration, Target 1 changes its heading by 0.25° counter-clockwise until it has a heading just north of 40°, at which point it starts turning clockwise at the same rate until it has a heading of 90°, at which point it turns counter-clockwise and repeats this sequence. Meanwhile, its velocity is decelerating at a rate of 10 km/h/sec until it reaches the minimum $\dot{x}$ or $\dot{y}$ (of $-495$ km/h), at which point it begins accelerating at a rate of 10 km/h/sec until it reaches the maximum $\dot{x}$ or $\dot{y}$ (495 km/h), at which point

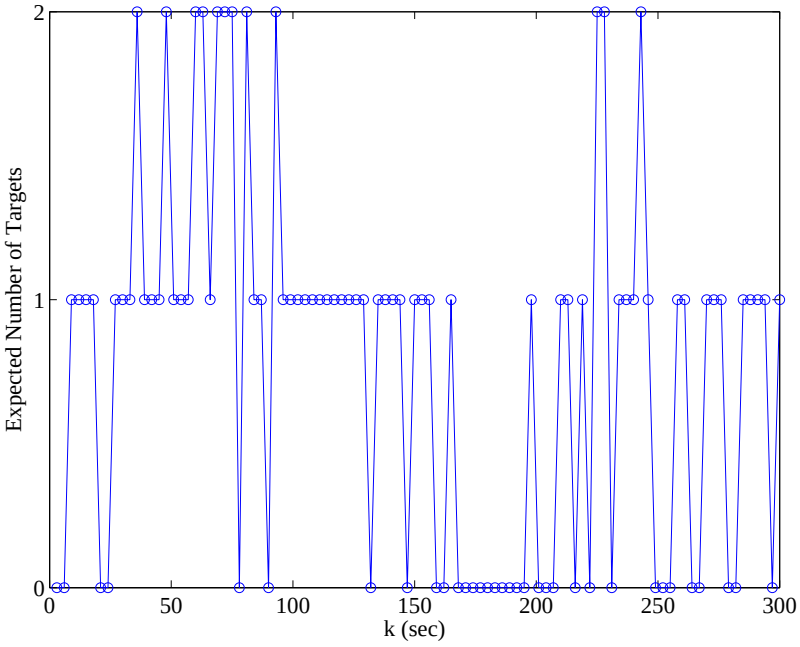



**Figure 64:** Expected number of targets at each time step in the two-target simulation with multipath effects using range, Doppler, and DOA observations to track.

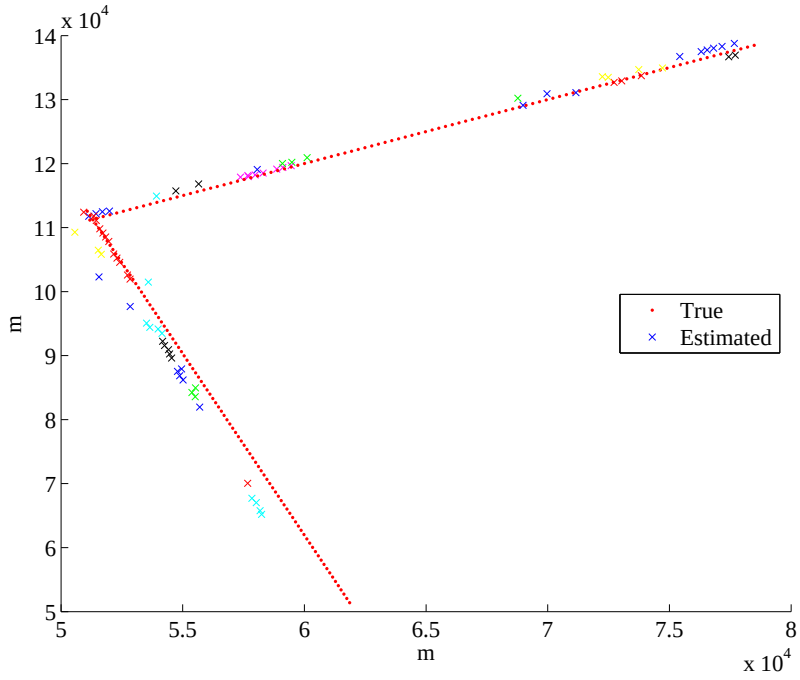
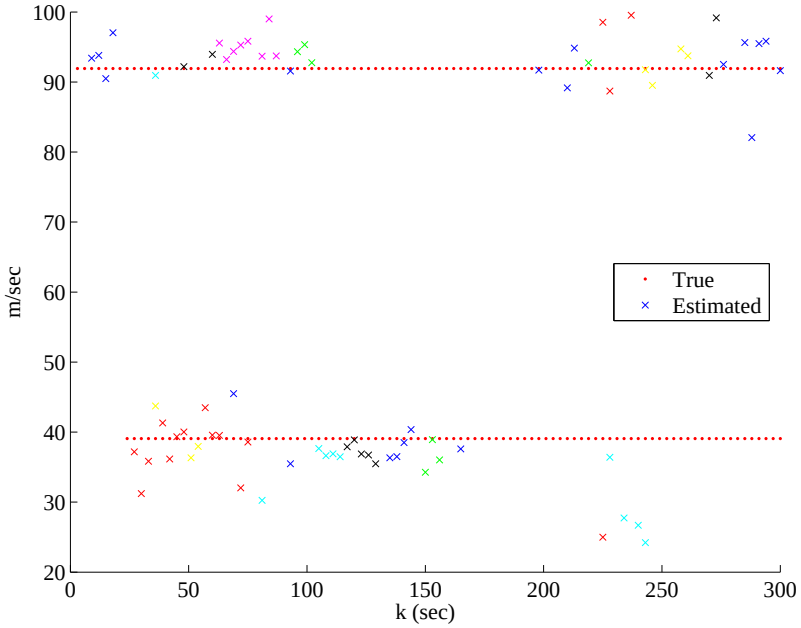
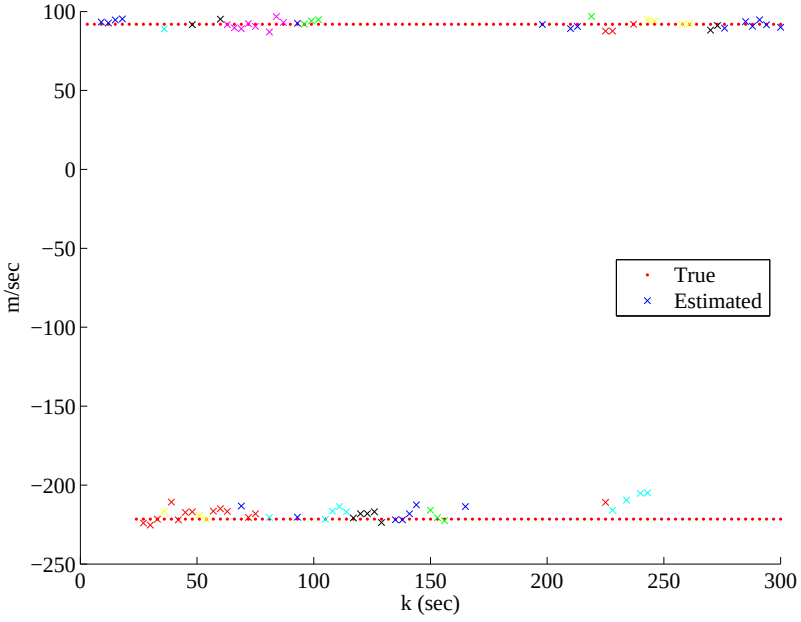



**Figure 65:** Actual vs. estimated target locations over the first 300 iterations of the two-target simulation with multipath effects using range, Doppler, and DOA observations to track.




(a) $\dot{x}$




(b) $\dot{y}$

**Figure 66:** Actual vs. estimated target velocities over the first 300 iterations of the two-target simulation with multipath effects using range, Doppler, and DOA observations to track.

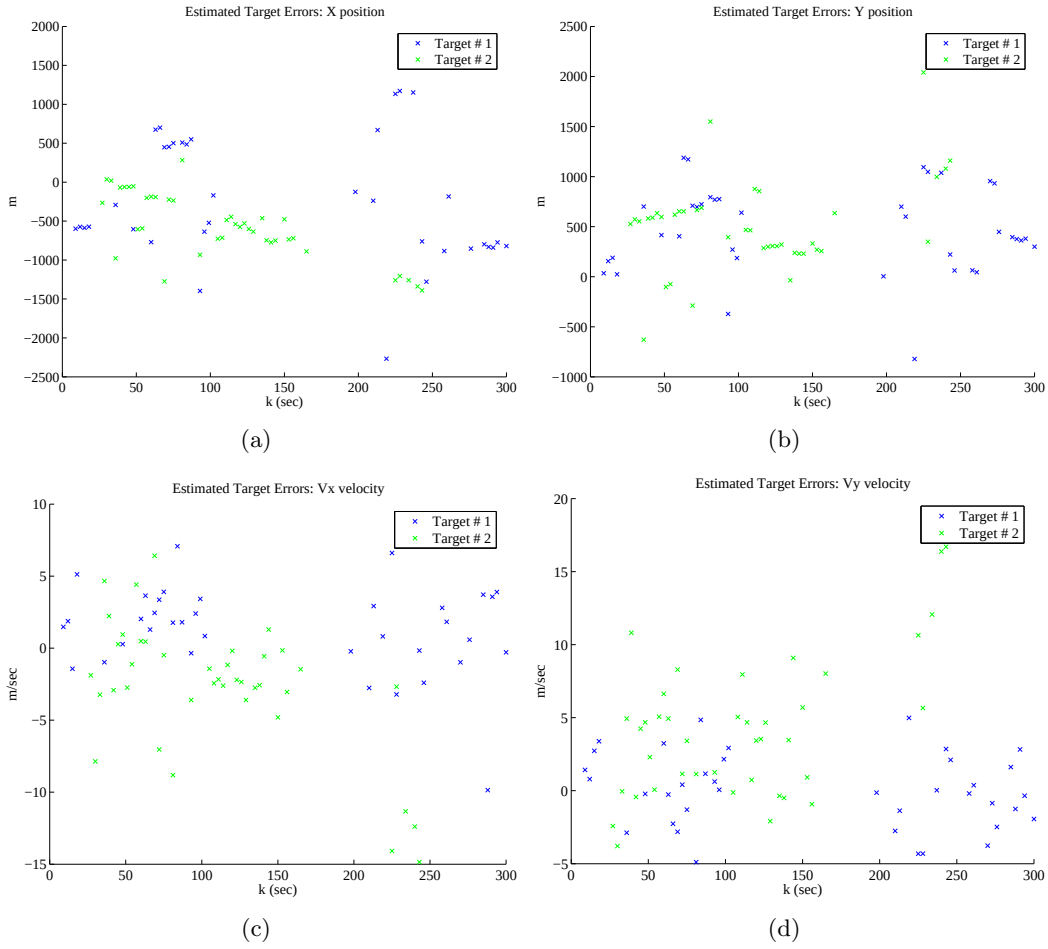



**Figure 67:** Errors in target state estimation by the PHD filter in the two-target simulation with multipath effects using range, Doppler, and DOA observations to track. The range resolution is around 6.7 km, and the Doppler resolution is a little over 3 m/sec.

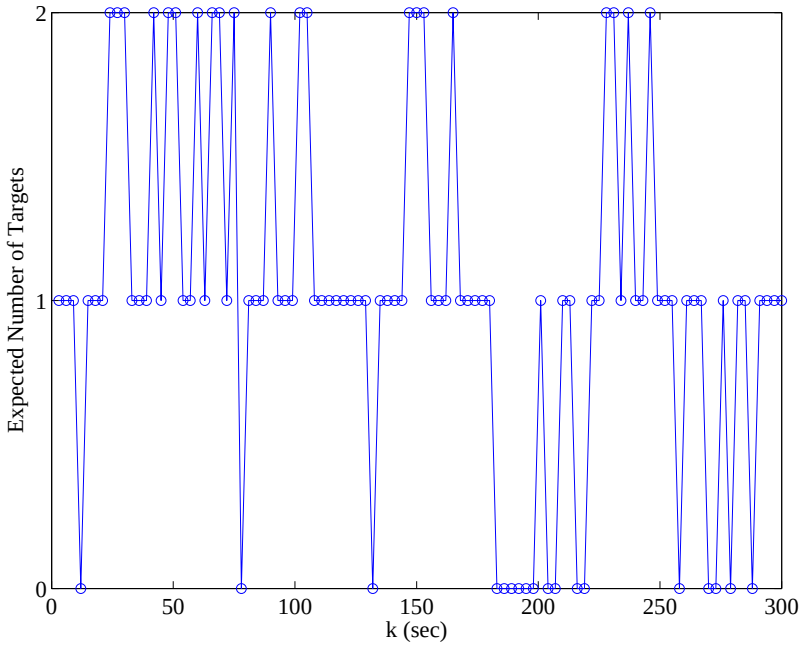



**Figure 68:** Expected number of targets at each time step in the two-target simulation with multipath effects using range, Doppler, and DOA observations to track. The number of particles used is a factor of 10 greater than that used to obtain Figure 64.

it begins to decelerate.

Figures 70-74 show the results of this scenario for a simulation with a range and velocity tracker and no multipath effects. Multipath effects are added, and the result is shown in Figures 75-79. Finally, DOA observations are included, and the effect is seen in Figures 80-83. As expected from Sections 6.4.3 and 6.5.5, the simulation without multipath effects or DOA observations performs the best. Next best, albeit with considerable target number overestimation, is the multipath-inclusive simulation without DOA observations. Note, however, that the poorest performing simulation of the three, the one with multipath and DOA observations, is the only one without ghost target detections.

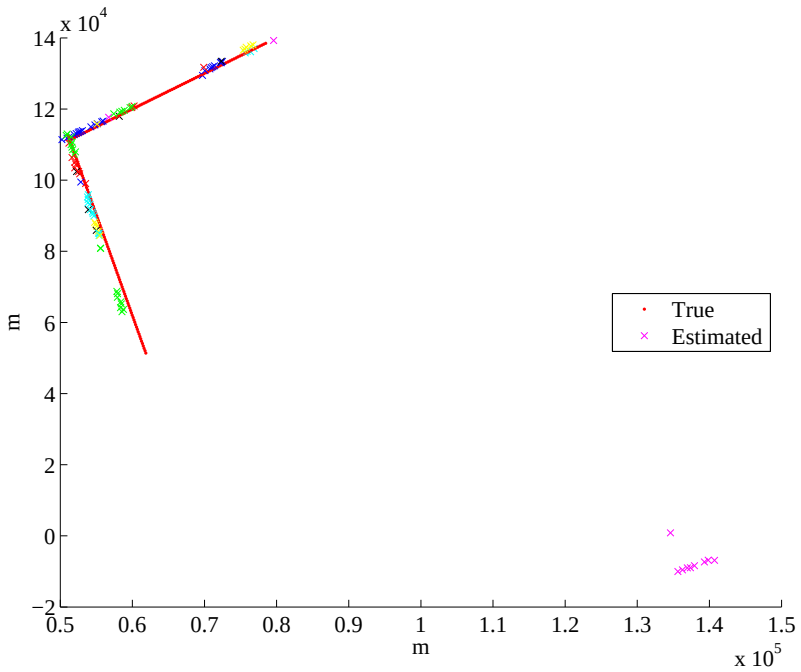



**Figure 69:** Actual vs. estimated target locations over the first 300 iterations of the two-target simulation with multipath effects using range, Doppler, and DOA observations to track. The number of particles used is a factor of 10 greater than that used to obtain Figure 65.

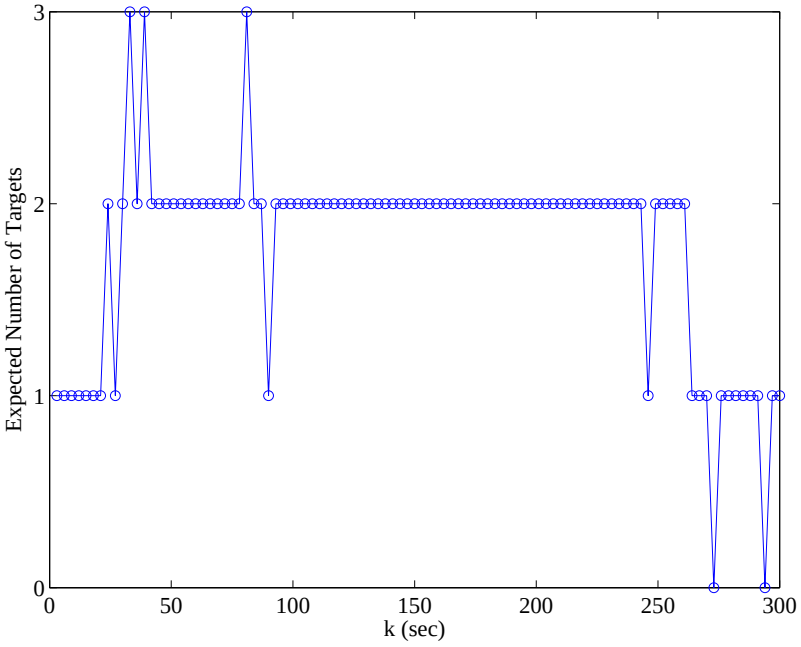



**Figure 70:** Expected number of targets at each time step in simulation with accelerating Target 1, no multipath, and range and Doppler observations only.

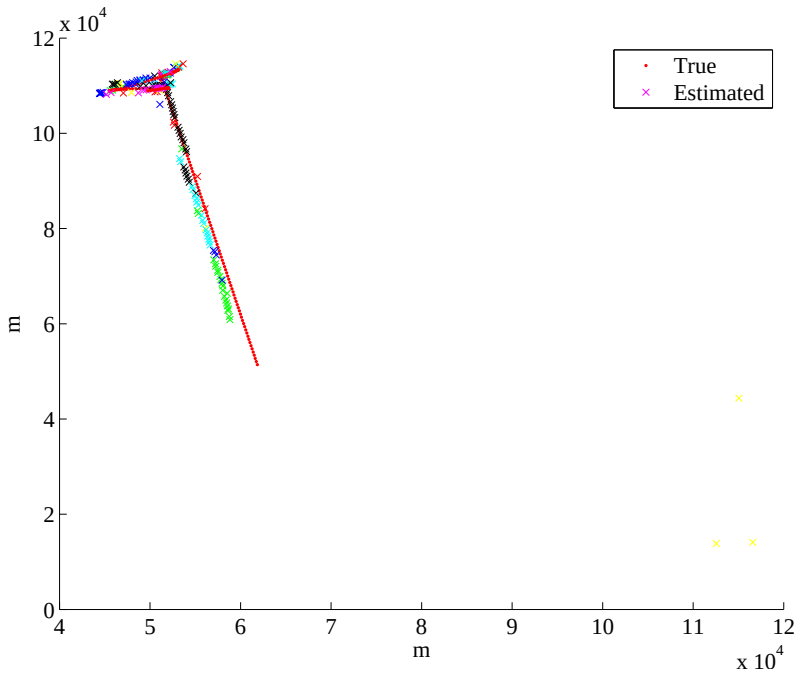



**Figure 71:** Actual vs. estimated target locations over the first 300 iterations of simulation with accelerating Target 1, no multipath, and range and Doppler observations only.

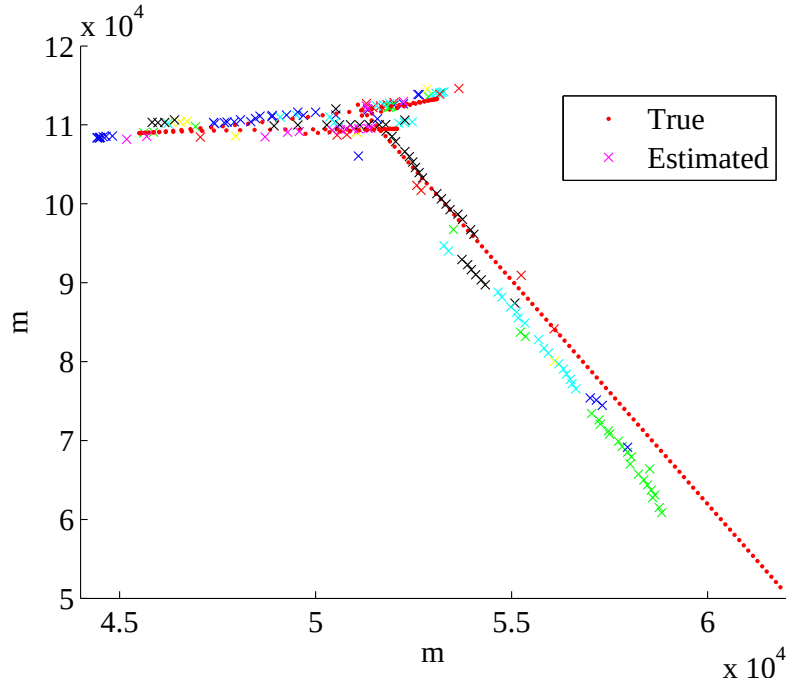
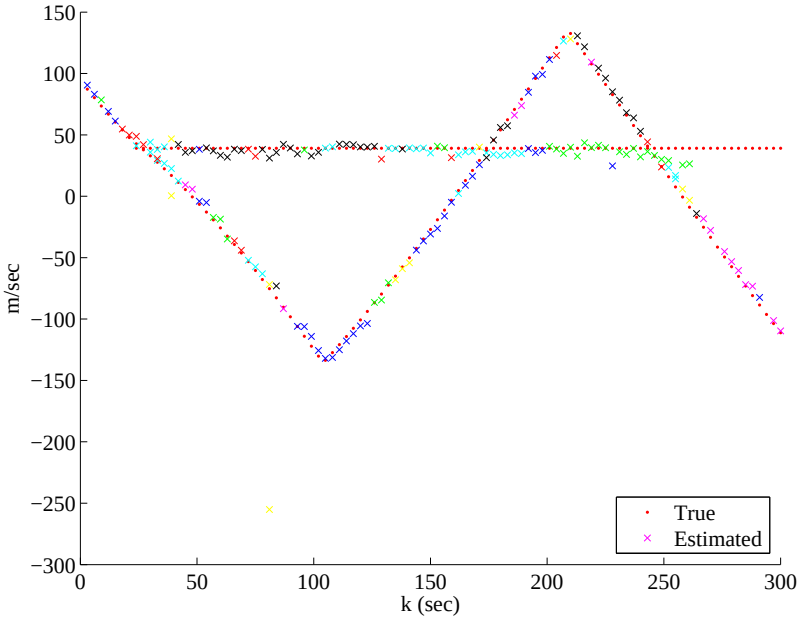
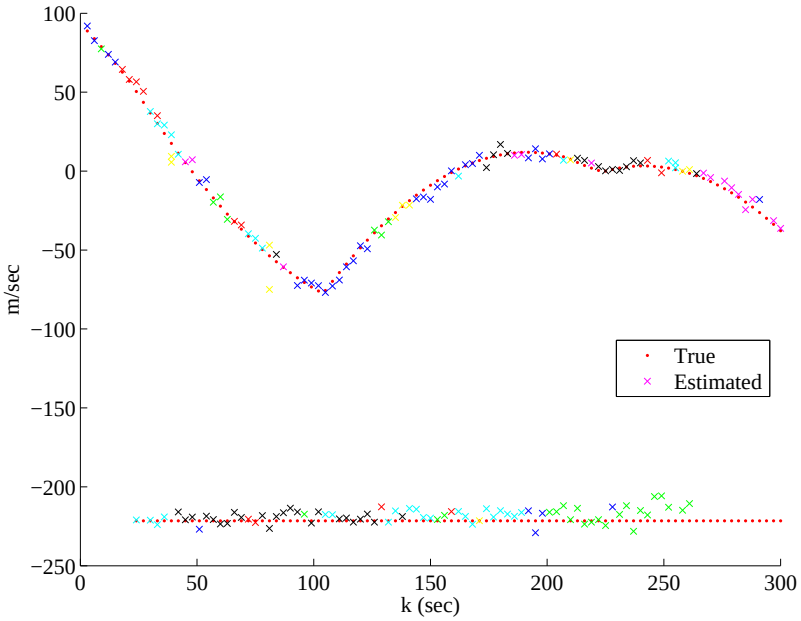



**Figure 72:** Close-up of actual vs. estimated target locations over the first 300 iterations of simulation with accelerating Target 1, no multipath, and range and Doppler observations only.




(a) $\dot{x}$




(b) $\dot{y}$

**Figure 73:** Actual vs. estimated target velocities over the first 300 iterations of simulation with accelerating Target 1, no multipath, and range and Doppler observations only.

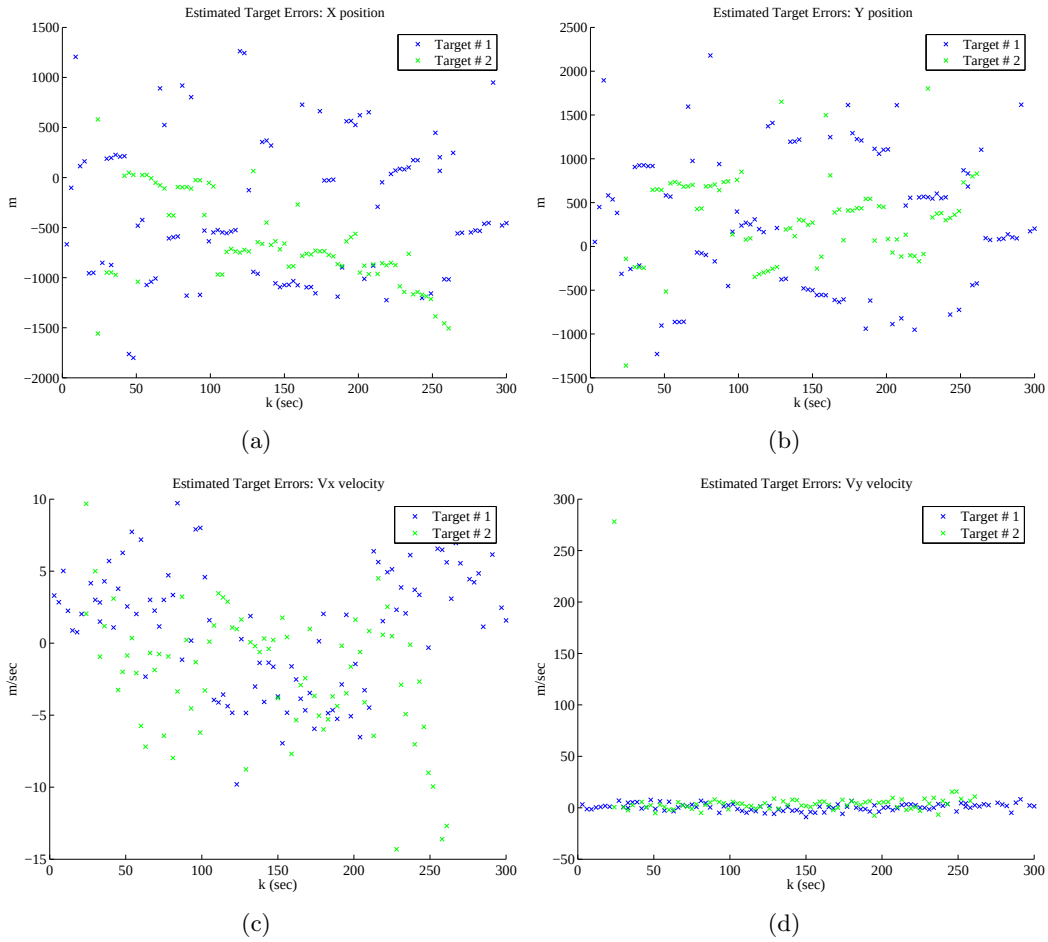



**Figure 74:** Errors in target state estimation by the PHD filter in the simulation with accelerating Target 1, no multipath, and range and Doppler observations only. The range resolution is around 6.7 km, and the Doppler resolution is a little over 3 m/sec.

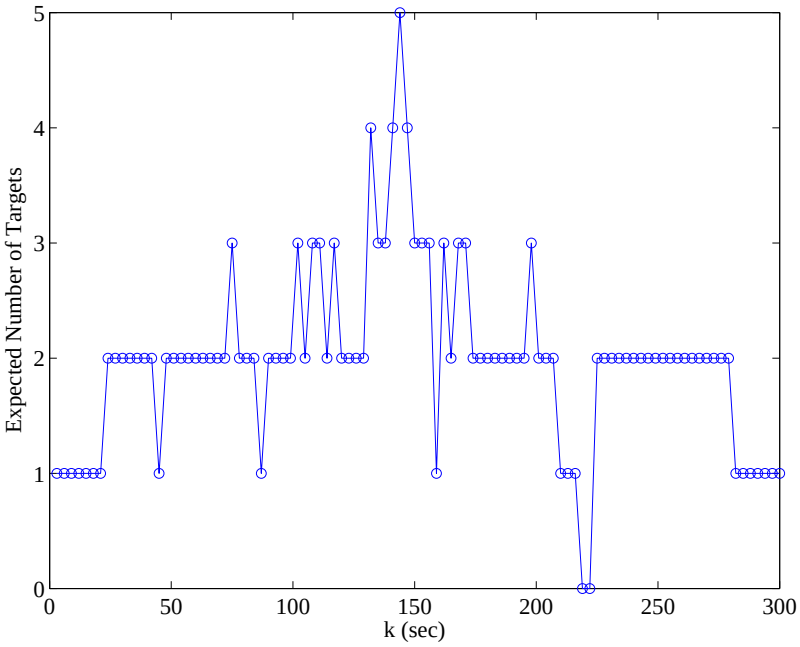



**Figure 75:** Expected number of targets at each time step in simulation with accelerating Target 1, multipath, and range and Doppler observations only.

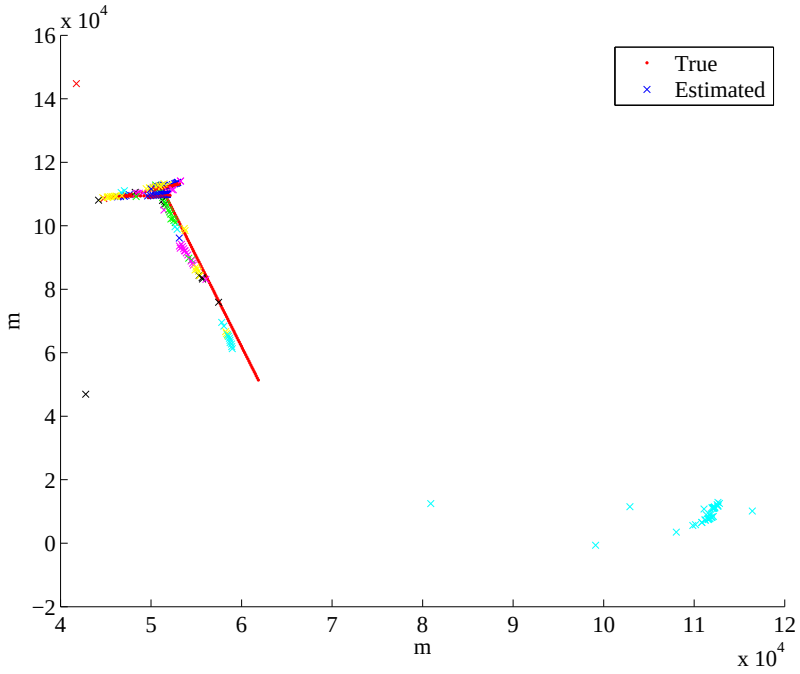



**Figure 76:** Actual vs. estimated target locations over the first 300 iterations of simulation with accelerating Target 1, multipath, and range and Doppler observations only.

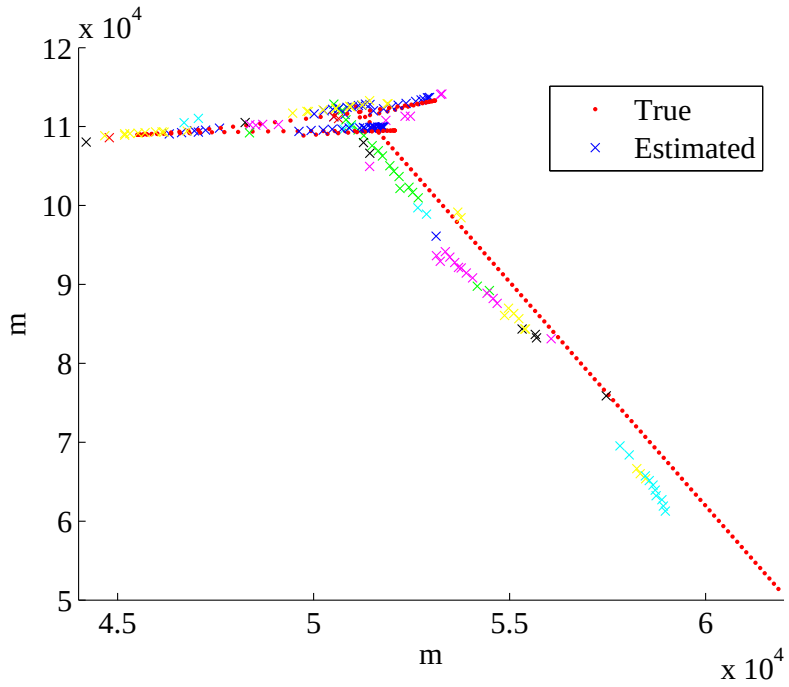
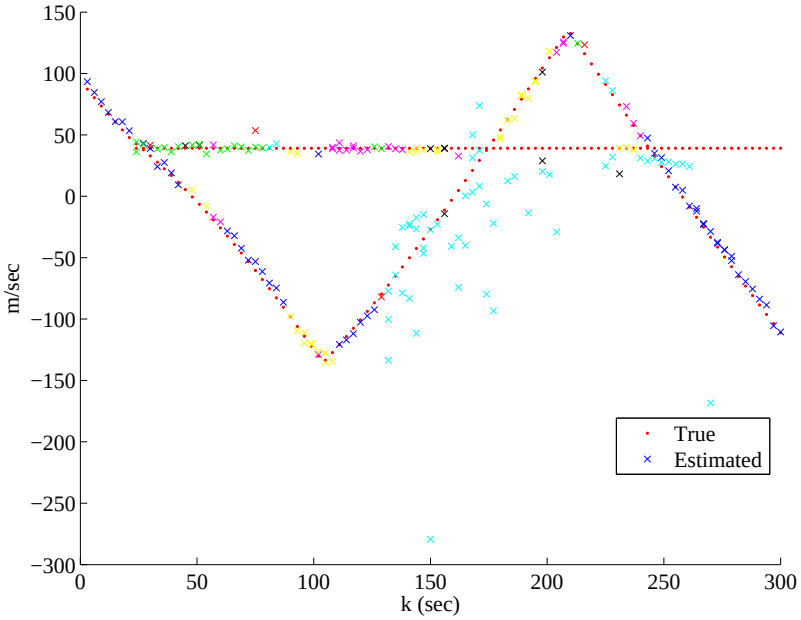
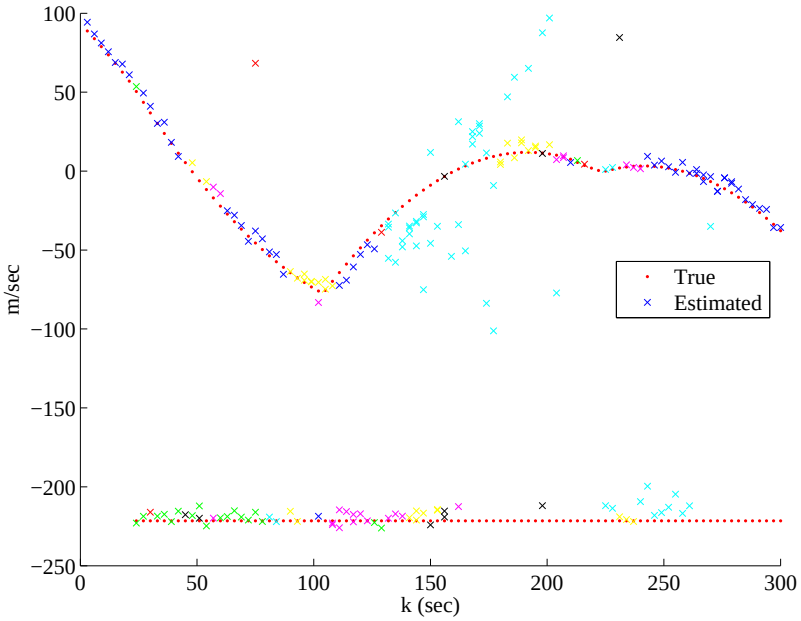



**Figure 77:** Close-up of actual vs. estimated target locations over the first 300 iterations of simulation with accelerating Target 1, multipath, and range and Doppler observations only.




(a) $\dot{x}$




(b) $\dot{y}$

**Figure 78:** Actual vs. estimated target velocities over the first 300 iterations of simulation with accelerating Target 1, multipath, and range and Doppler observations only.

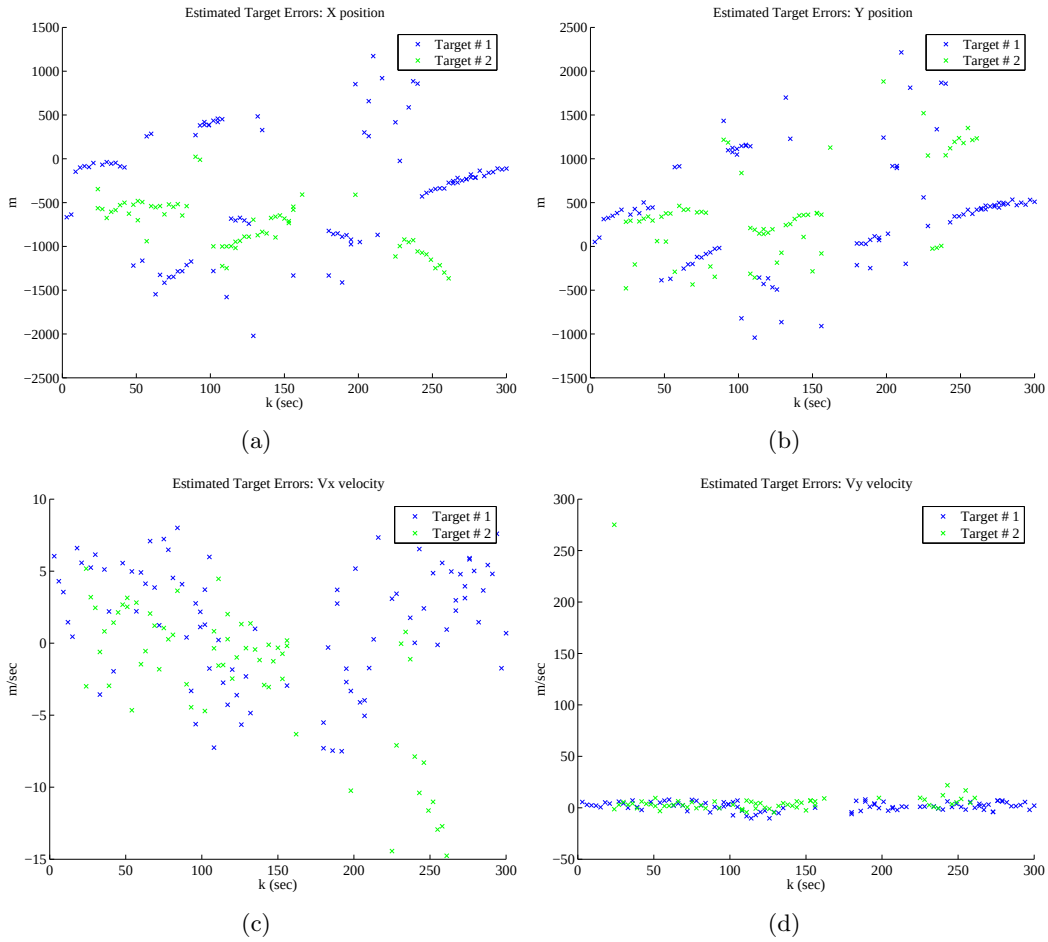



**Figure 79:** Errors in target state estimation by the PHD filter in the simulation with accelerating Target 1, multipath, and range and Doppler observations only. The range resolution is around 6.7 km, and the Doppler resolution is a little over 3 m/sec.

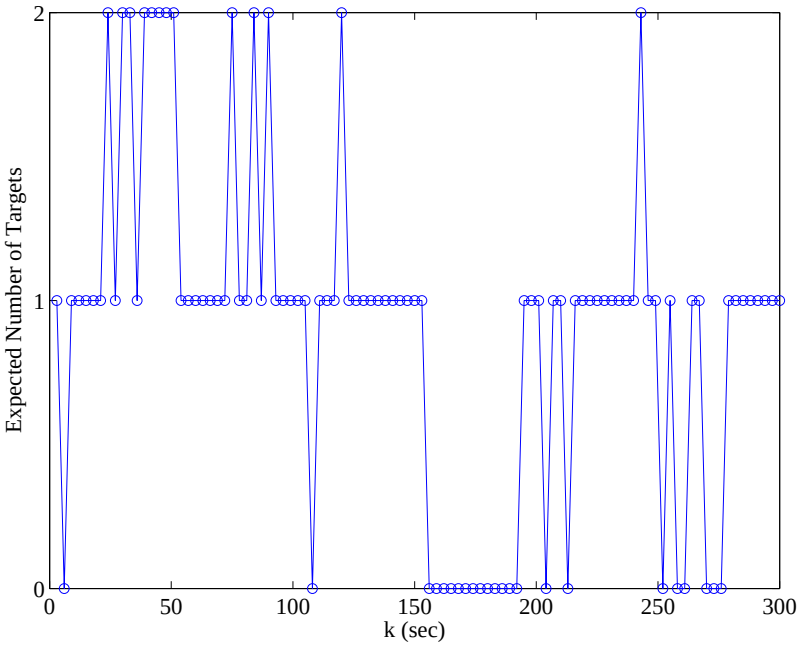



**Figure 80:** Expected number of targets at each time step in simulation with accelerating Target 1, multipath, and range, Doppler, and DOA observations.

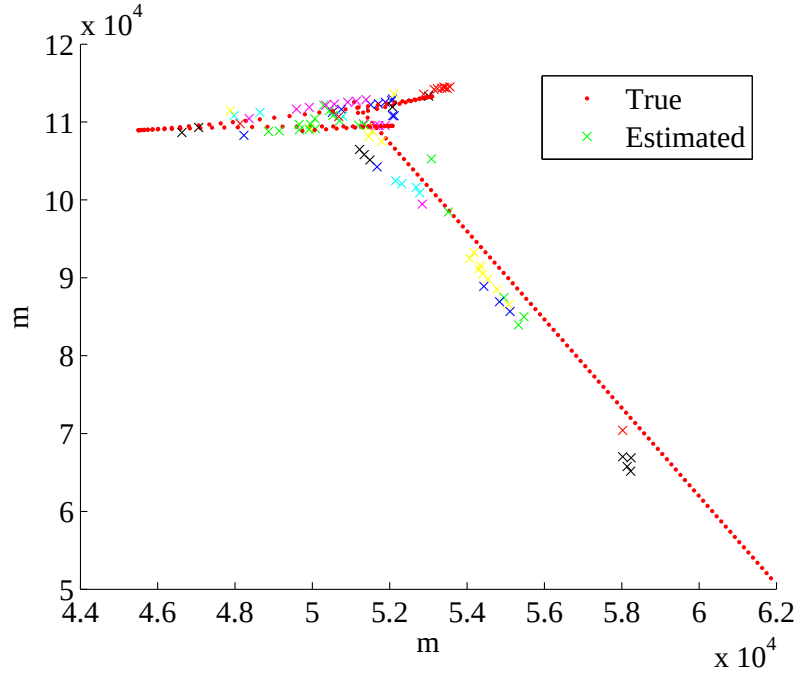
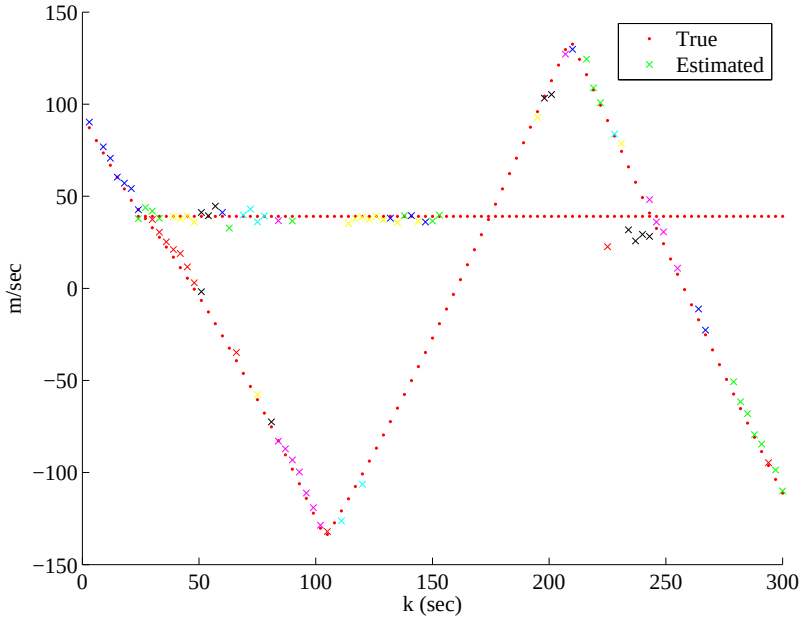
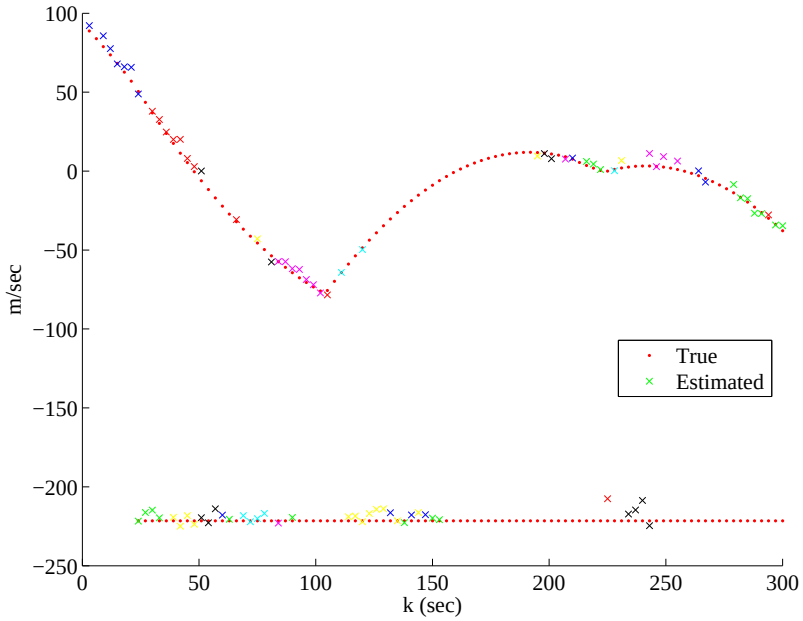



**Figure 81:** Actual vs. estimated target locations over the first 300 iterations of simulation with accelerating Target 1, multipath, and range, Doppler, and DOA observations.




(a) $\dot{x}$




(b) $\dot{y}$

**Figure 82:** Actual vs. estimated target velocities over the first 300 iterations of simulation with accelerating Target 1, multipath, and range, Doppler, and DOA observations.

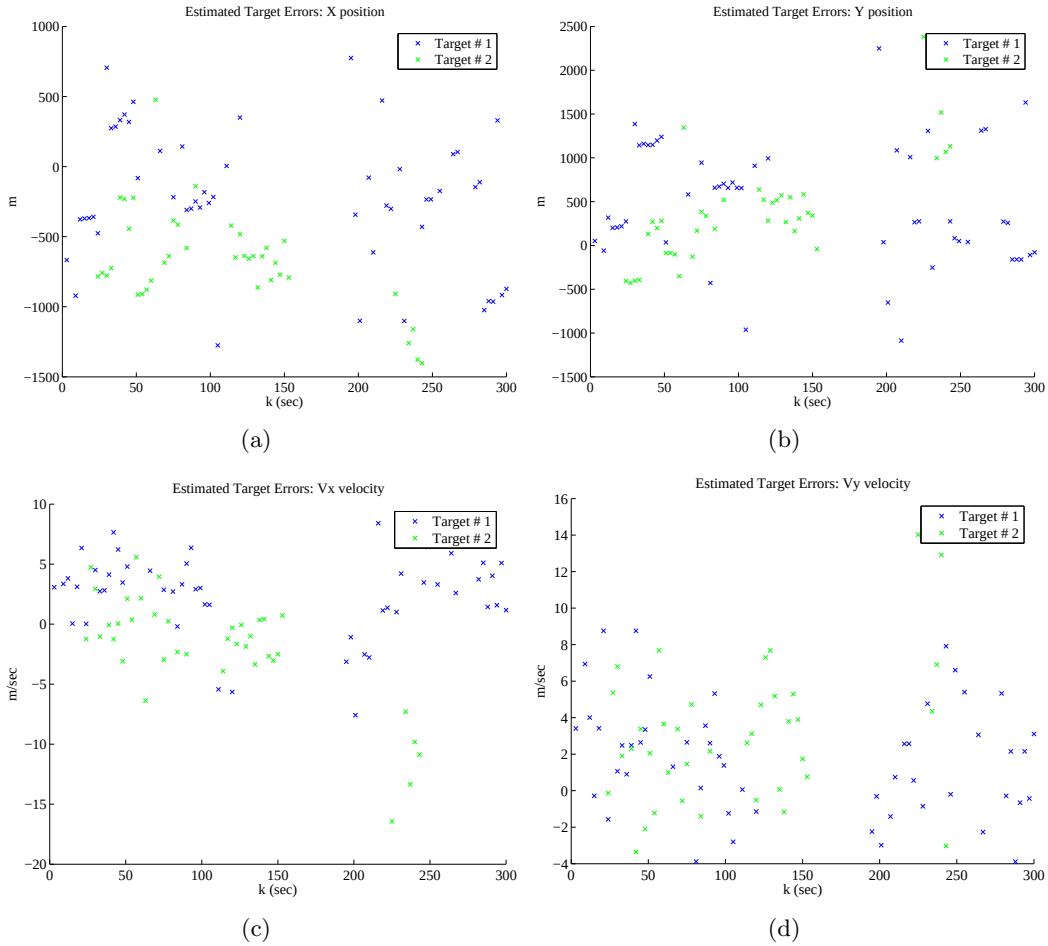



**Figure 83:** Errors in target state estimation by the PHD filter in the simulation with accelerating Target 1, multipath, and range, Doppler, and DOA observations. The range resolution is around 6.7 km, and the Doppler resolution is a little over 3 m/sec.

# CHAPTER VII

# CONCLUSIONS

Our initial foray into the use of the probability hypothesis density (PHD) for multitarget, multisensor tracking has been insightful. The realistic passive-radar simulations described in Chapters 4-6 demonstrate that the PHD is a promising tool for estimating number of targets, detecting multiple target states, and easily fusing different kinds of observation data from multiple sensors. By itself, however, the PHD is not a complete target tracking solution, since it lacks the necessary track association functionality. This section summarizes some of the limitations of the PHD particle filter and suggests possible directions of future work.

## *7.1 Weaknesses in the PHD Filter: Track Association and Likelihood Model Sensitivity*

As discussed in section 6.2.1, our attempt at adding explicit tracking functionality to the PHD by inserting a track ID into the particle states was unsuccessful. Better results might be obtained by using a smarter particle placement algorithm for the particles propagating from one time step to the next. However, we may be trying to add a feature to the PHD for which it is not designed to handle. The PHD filter uses a Markov motion model to predict where targets will be from one time step to the next, but it does not keep track of the identity of the targets in the process. The random set theory used by FISST and the PHD provides a useful tool merely to analyze the existence of objects, such as targets or sensor observations in regions of state space or observation space. Unlike a vector representation of a set of objects, the random set does not care in which order the objects are placed. The result is that given a random set of targets $X_k = \{\mathbf{x}_{1,k}, \mathbf{x}_{2,k}\}$ at time $k$ and a random set $X_{k+1} = \{\mathbf{x}_{1,k+1}, \mathbf{x}_{2,k+1}\}$ at time $k+1$, there is no association between $\mathbf{x}_{1,k}$ and $\mathbf{x}_{1,k+1}$, since the position of these objects in the random sets are arbitrary. This presents a limitation

for the PHD in terms of target tracking, since there is no target association from one moment in time to the next. Any such functionality might best be left to a post-processing algorithm that uses the PHD peak-extractions to form tracks. Thus, to be of practical use for multitarget tracking, the PHD requires post-processing by a nearest-neighbor algorithm or other more sophisticated techniques, such as have been proposed in [37] and [62].

The main benefit of the PHD, however, is not in its tracking ability, but rather in the ease at which it handles data fusion. It fuses a large quantity of data and reports the result naturally in a cartesian coordinate system. Thus, in a multitarget tracker implementation, the PHD could best be used as an initial filter that fuses data from multiple sensors and feeds the results, as if from a single sensor, to a post-processing filter.

The advantage of the PHD filter is that it allows one to fuse data easily via the single target motion likelihoods and single sensor likelihoods. However, this presents a problem when there are targets with different motion models. Since the PHD filter does not distinguish between targets (i.e., no track association is performed), there is no automatic way for the PHD filter to choose which motion model to apply during the prediction step of the Bayes filter. In the Time Update step, (83), the probability of survival and the target motion model both depend on the target state $\mathbf{x}$, but since the target state does not in itself currently provide any target-identifying information, the choice of which target motion model (and, possibly, probability of survival model) to use is not clear. A potential solution to this problem is presented in [65], which incorporates multiple models into the PHD filter.

Unfortunately, the results of the multipath simulations (see Section 6.4.3) indicate that the performance of the PHD filter is highly dependent on the accuracy of the filter parameters to matching those in the real scenario. Having the correct signal-to-noise ratio and probability of detection are essential. If the probability of detection is modeled too high, then a target is quickly dropped upon entering a region of low $p_D$. This issue, even when the correct $p_D$ is used, is under investigation by Erdinc *et al.* [19], as mentioned in Section 6.4.2. If too low a $p_D$ is assumed, then overestimation of the number of targets occurs, because the PHD filter assumes it cannot observe the targets, even though it does. In tandem with an incorrect $p_D$, if the SNR value is incorrect, then the sensor likelihood function fails to

correctly model the sensor behavior. If the SNR is assumed to be higher than it really is, this leads to failure in target detection. In the case where the PHD filter assumes too low an SNR, this leads to target-number overestimation, even in the absence of false alarms.

The best performance in the multipath simulation experiments was obtained when the correct SNR and $p_D$ were known by the PHD filter to determine the correct sensor likelihoods in the Data Update stage.[1] We suggest that by adding $z$ and $\dot{z}$ components to the particles, the Data Update step will have available to it probable altitude values for the targets, and hence will be able to determine appropriate SNR and $p_D$ values. However, the altitude information that can be obtained from a PCL system is not as good as the $x$ and $y$ information. Whether the range-variance based grid was constructed knowing the true multipath effect or not was shown to have little effect on the performance of the tracker. This is fortunate, since the grid used for finding observation intersections is too inflexible to change in real time to incorporate changing target altitudes. To do so might require a three-dimensional version of the iterative least-squares algorithm, introduced in Section 5.3.5, for finding the range- and Doppler-observation intersections. The grid is also only practical if the transmitters and receivers are immobile. Were they to move, the grid would require recalculation, and this would not be possible in real time. The iterative least-squares algorithm may prove to be a better technique to use in this case.

## *7.2 Avenues of Future Research*

The need for smarter proposal functions to place particles has been a recurring theme throughout this research. As demonstrated in the maneuvering target simulations of Section 6.6, the current PHD particle-filter implementation can track a target with changing velocity. However, the performance may be improved by using a better model for propagating particles in the Time Update step. A method of incorporating observation information to create a Gaussian-distribution from which to sample the propagating particles is described in [61]. However, applying this approach in the case of multiple observations and

[1] Even then, Target 2 could not be located during the period where the probabilities of detection from the transmitters used were insufficient.

missed detections requires further research.

The Data Update step could also benefit from a smarter birth placement algorithm. As noted in Section 6.5.5, incorporating the DOA observations into the birth particle placement method may lead to improved target detection when using range, Doppler, and DOA sensor measurements. More complicated DOA observations, such as from implementing beamforming at the receiver, would also lead to better performance. As discovered when adding velocity measurements into the range-only tracker, the PHD filter performs data fusion with ease, and with the introduction of additional high-quality observations, the target detection performance improves. It is thus worthwhile to investigate smarter particle placement techniques, so that the non-optimality of the particle filter does not prevent the target detection capability of the PHD filter from improving when additional data is introduced.

The grid method, presented in Chapter 5, for the smart placement of birth particles is attractive in its accuracy at localizing ellipse intersections and in its ability to be precomputed before the simulation is run. However, in the case of moving transmitters or receiver, the grid would need to be re-computed at each time step, rendering it impractical for real-time use. The alternative least-squares iterative approach would be more flexible, but it also requires a good starting point, which necessitates additional target information. Both methods are of combinatorial complexity and are limited in the number of ellipse observations they can process in real-time. Developing an alternative method of finding the ellipse intersections, or instead, an alternative method of cleverly placing the birth particles, may be an avenue for future work. In addition, allowing the grid method to place particles in those gridspaces that contain intersections of fewer than three ellipses (from different transmitters) is an avenue for future work. This would allow the PHD particle filter to place birth particles at target locations where the probabilities of detection are not sufficiently high enough from all of the transmitters.

In addition to the functional improvements to the PHD particle filter already mentioned, the application of the PHD filter to various scenarios should be further expanded and tested. Situations involving spawning targets, as well as varying probabilities of survival, should be tested. Furthermore, comparison of the PHD with other types of target tracking

methods is also worthy of considerable research. A comparison between the PHD and multiple hypothesis tracking has been performed in [62]. An extended Kalman filter, or other conventional tracker, with appropriate data association should be applied to the same multitarget, passive coherent location (PCL) scenarios, so that a comparison may be made between the performance of the PHD filter and that of other trackers in the context of passive radar.

Further research into FISST and the PHD is also possible. The PHD is analogous to the mean used in the Kalman filter in single-target tracking. It would be nice to obtain a second-moment, analagous to the covariance matrix in the Kalman filter, to propagate along with the PHD. Mahler attempted to do so, but he concluded that it is intractable and proposed a theory of the "multitarget extended Kalman Filter" as an alternative [46, 50].

## *7.3 Contributions*

To demonstrate the functionality of the PHD and analyze its performance, we have applied it to multitarget, multisensor tracking using passive radar. In the absence of real data, we have attempted to be as realistic as possible in our simulations, and we believe that we are the first to stress the PHD-based tracker in a non-idealized multitarget, multisensor scenario with both false alarms and realistic, fluctuating probabilities of detection.

We have also used range, Doppler, and DOA observations to stress the target detection and data fusion capabilities of the PHD filter. With the improvements suggested in Sections 7.1 and 7.2, the PHD filter appears to be a solution to the data-fusion problem. We have demonstrated that the PHD filter would make a promising tool for tracking multiple targets with passive radar, but that by itself, it is not a complete tracking solution. To be so, the PHD filter would require a post-processor for handling target-track formation and maintenance.

To improve the performance of the PHD particle filter, we have devised ellipse-intersection logic that takes advantage of bistatic range and Doppler observations to propose smarter densities for particle placement. A pre-computable grid technique both improved the target detection performance and greatly reduced the number of particles needed in the filter.

We have also designed and implemented a novel peak-extraction technique to locate the peaks in the PHD. The technique takes advantage of the property of the PHD particle filter that the particle weights integrate to the number of targets present in any given area. Compared to the EM and $k$-means algorithms, our new peak-extraction technique is simpler and more effective. It is considerably faster than the EM algorithm, as well.

It is also hoped that the probability hypothesis density and the PHD filter, as well as their FISST underpinnings, were presented in an easily understandable fashion, and that the reader of this thesis has gained a better comprehension of FISST and the PHD and of the strengths and weaknesses of the Bayesian PHD particle filter for multitarget, multisensor tracking.

# REFERENCES


[1] Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T., "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. on Signal Proc.*, vol. 50, no. 2, pp. 174–188, 2002.

[2] Bar-Shalom, Y. and Li, X.-R., *Multitarget-Multisensor Tracking: Principles and Techniques*. YBS Publishing, 1995.

[3] Barton, D., *Modern Radar System Analysis*. Artech House, Inc., 1988.

[4] Biernacki, C., Celeux, G., and Govaert, G., "Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate gaussian mixture models," *Computational Statistics and Data Analysis*, vol. 41, pp. 561–575, Jan. 2003.

[5] Bilmes, J. A., "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden markov models," Tech. Rep. TR-97-021, International Computer Science Institute, Berkeley, CA, April 1998.

[6] Blackman, S. S., *Multiple Target Tracking with Radar Applications*. Artech House, 1986.

[7] Blom, H. A. P. and Bar-Shalom, Y., "The interacting multiple model algorithm for systems with markovian switching coefficients," *IEEE Trans. on Automatic Control*, vol. 33, pp. 780–783, Aug. 1988.

[8] Blom, H. A. P. and Bloem, E. A., "Joint IMMPDA particle filter," *Proc. of the 6th Int'l Conf. of Info. Fusion*, vol. 2, pp. 785–792, 8-11 July 2003.

[9] Bose, R., Freedman, A., and Steinberg, B. D., "Sequence clean: A modified deconvolution technique for microwave images of contiguous targets," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 38, pp. 89–97, Jan. 2002.

[10] Cevher, V., *A Bayesian Framework for Target Tracking using Acoustic and Image Measurements*. PhD thesis, Georgia Institute of Technology, Atlanta, Georgia, 2005.

[11] Chong, C. Y., Mori, S., and Chang, K. C., "Distributed multitarget multisensor tracking," in *Multitarget-Multisensor Tracking: Advanced Applications* (Bar-Shalom, Y., ed.), ch. 8, Artech House, 1990.

[12] Clark, D. and Bell, J., "Bayesian multiple target tracking in forward scan sonar images using the PHD filter," *IEE Proc. Radar, Sonar and Navigation*, vol. 152, pp. 327–334, October 2005.

[13] Clark, D., Bell, J., de Saint-Pern, Y., and Petillot, Y., "PHD filter multi-target tracking in 3D sonar," in *Oceans 2005 - Europe*, vol. 1, (Brest, France), pp. 265–270, 20-23 June 2005.

[14] Daley, D. and Vere-Jones, D., *An Introduction to the Theory of Point Processes*, vol. 1. Springer-Verlag, 2 ed., 2003.

[15] Doucet, A., de Freitas, N., and Gordon, N., *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.

[16] Doucet, A., Godsill, S., and Andrieu, C., "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, pp. 197–208, July 2000.

[17] Ehrman, L. M., "Automated target recognition using passive radar and a coordinated flight model," Master's thesis, Georgia Institute of Technology, Atlanta, Georgia, May 2004.

[18] El-Fallah, A., Zajic, T., Mahler, R., Lajza-Rooks, B., and Mehra, R., "Multitarget nonlinear filtering based on spectral compression and probability hypothesis density," in *Signal Processing, Sensor Fusion, and Target Recognition X* (Kadar, I., ed.), vol. Proc. SPIE 4380, pp. 207–216, 2001.

[19] Erdinc, O., Willett, P., and Bar-Shalom, Y., "Probability hypothesis density filter for multitarget multisensor tracking," in *Proc. 8th International Conf. on Information Fusion*, (Philadelphia, Pennsylvania), July 2005.

[20] Figueiredo, M. A. T., Leitao, J. M. N., and Jain, A. K., "On fitting mixture models," in *Energy Minimization Methods in Computer Vision and Pattern Recognition* (Hancock, E. and Pellilo, M., eds.), pp. 54–69, Springer-Verlag, 1999.

[21] Goodman, I. R., Mahler, R. P. S., and Nguyen, H. T., *Mathematics of Data Fusion*. Kluwer Academic Publishers, 1997.

[22] Hatch, M. D., Kaina, J. L., Mahler, R. P., and Myre, R. S., "Data fusion methodologies to support theater level and deployable surveillance systems," in *Conf. Record of the Thirty-Second Asilomar Conference on Signals, Systems, and Computers*, vol. 1, pp. 563–567, 1998.

[23] Herman, S. C. and Moulin, P., "A particle filtering approach to FM-band passive radar tracking and automatic target recognition," in *Proc. IEEE Aerospace Conference*, vol. 4, (Big Sky, Montana), pp. 1789–1808, March 9-16 2002.

[24] Hoffman, J., Mahler, R., Ravichandran, R., Huff, M., and Musick, S., "Robust SAR ATR by hedging against uncertainty," in *Signal Processing, Sensor Fusion, and Target Recognition XI* (Kadar, I., ed.), vol. SPIE Proc. 4729, pp. 187–198, 2002.

[25] Hoffman, J. and Mahler, R., "Multitarget miss distance and its applications," in *Proc. Fifth International Conference on Information Fusion*, pp. 149–155, 2002.

[26] Howland, P. E., *Television Based Bistatic Radar*. PhD thesis, University of Birmingham, ENGLAND, 1997.

[27] Howland, P. E., "Target tracking using television-based bistatic radar," *IEE Proc. F: Radar, Sonar, and Navigation*, vol. 146, pp. 166–174, June 1999.

[28] Howland, P. E., Maksimiuk, D., and Reitsma, G., "FM radio based bistatic radar," *IEE Proc. Radar, Sonar, and Navigation*, vol. 152, pp. 107–115, June 2005.

[29] Huff, M., Yu, S.-H., Mahler, R., Ravichandran, B., Mehra, R., and Musick, S., "Unified evidence accrual for SAR: Recent results," in *Signal Processing, Sensor Fusion, and Target Recognition IX* (Kadar, I., ed.), vol. SPIE Proc. 4052, pp. 149–159, 2000.

[30] Isard, M. and Blake, A., "ICondensation: Unifying low-level and high-level tracking in a stochastic framework," in *Proc. 5th European Conf. Computer Vision*, vol. 1, pp. 893–908, 1998.

[31] Isard, M. and Blake, A., "Partitioned sampling, articulated objects, and interface-quality hand tracking," in *Proc. European Conf. Computer Vision*, vol. 2, pp. 3–19, 2000.

[32] Johansen, A. M., Singh, S. S., Doucet, A., and Vo, B.-N., "Convergence of the SMC implementation of the PHD filter," Technical Report CUED/F-INFENG/TR-517, University of Cambridge, April 2005.

[33] Kalman, R., "A new approach to linear filtering and prediction problems," *Trans. of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[34] Kamen, E. W., "Multiple target tracking based on symmetric measurement equations," *IEEE Trans. on Automatic Control*, vol. 37, pp. 371–374, April 1992.

[35] Kamen, E. W. and Sastry, C. R., "Multiple target tracking using products of position measurements," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 29, no. 2, pp. 476–493, 1993.

[36] Kamen, E. W. and Su, J., *Introduction to Optimal Estimation*. Springer Verlag, 1999.

[37] Lin, L., Bar-Shalom, Y., and Kirubarajan, T., "Data association combined with the probability hypothesis density filter for multitarget tracking," in *Signal and Data Processing of Small Targets 2004* (Drummond, O. E., ed.), vol. Proc. SPIE 5428, April 2004.

[38] Liu, J. S. and Chen, R., "Sequential monte carlo methods for dynamic systems," *Amer. Statistical Assoc.*, vol. 93, no. 443, pp. 1032–1044, 1998.

[39] Mahafza, B., *Introduction to Radar Analysis*. CRC Press LLC, 1998.

[40] Mahler, R., "Nonadditive probability, finite-set statistics, and information fusion," in *Proc. of the 34th Conf. on Decision and Control*, (New Orleans, LA), pp. 1947–1952, Dec. 1995.

[41] Mahler, R., "Global posterior densities for sensor management," in *Acquisition, Tracking, and Pointing XII* (Master, M. and Stockum, L., eds.), vol. SPIE Proc. 3365, pp. 252–263, 1998.

[42] Mahler, R., "Practical information-based data fusion performance evaluation," in *Signal Processing, Sensor Fusion, and Target Recognition VII* (Kadar, I., ed.), vol. SPIE Proc. 3720, pp. 92–103, April 1999.

[43] Mahler, R., "An introduction to multisource-multitarget statistics and its applications," tech. rep., Lockheed Martin, 2000.

[44] Mahler, R., "Optimal/robust distributed data fusion: A unified approach," in *Signal Processing, Sensor Fusion, and Target Recognition IX* (Kadar, I., ed.), vol. SPIE Proc. 4052, pp. 128–138, 2000.

[45] Mahler, R., "Engineering statistics for multi-object tracking," in *IEEE Workshop on Multi-Object Tracking*, pp. 53–60, 2001.

[46] Mahler, R., "Multitarget moments and their application to multitarget tracking," in *Proc. of The Workshop on Estimation, Tracking, and Fusion: A Tribute to Yaakov Bar-Shalom*, (Monterey, CA), pp. 134–166, May 2001.

[47] Mahler, R., "'Statistics 101' for multisource-multitarget problems: Motivations, concepts, procedures, and applications." Tutorial. 5th International Conf. on Info. Fusion, July 2002.

[48] Mahler, R., "Objective functions for Bayesian control-theoretic sensor management, I: Multitarget first-moment approximation," in *Proc. of IEEE Aerospace Conference*, vol. 4, (Big Sky, MT), pp. 1905–1924, March 2003.

[49] Mahler, R., "Random sets: Unification and computation for information fusion – a retrospective assessment," in *Proc. of the 7th International Conference on Info. Fusion*, (Stockholm, Sweden), pp. 1–20, 28 June - 1 July 2004.

[50] Mahler, R., "A general theory of multitarget extended Kalman filters," in *Signal Processing, Sensor Fusion, and Target Recognition XIV* (Kadar, I., ed.), vol. Proc. SPIE 5809, (Orlando, Florida), pp. 208–219, 2005.

[51] Mahler, R. and O'Hely, M., "Multitarget detection and acquisition: A unified approach," in *Signal and Data Processing of Small Targets* (Drummond, O., ed.), vol. SPIE Proc. 3809, (Denver, CO), pp. 218–229, 1999.

[52] Mahler, R. and Prasanth, R., "Technologies for the unified collection and control of UAVs," in *Signal Processing, Sensor Fusion, and Target Recognition XI* (Kadar, I., ed.), vol. SPIE Proc. 4729, pp. 90–101, 2002.

[53] Mahler, R. and Zajic, T., "Multitarget filtering using a multitarget first-order moment statistic," in *Signal Processing, Sensor Fusion, and Target Recognition X* (Kadar, I., ed.), vol. Proc. SPIE 4380, pp. 184–195, 2001.

[54] Mahler, R. P., "Detecting, tracking, and classifying group targets: A unified approach," in *Signal Processing, Sensor Fusion, and Target Recognition X* (Kadar, I., ed.), vol. SPIE Proc. 4380, pp. 217–228, 2001.

[55] Mahler, R. P. S., "Search for tractable Bayesian multitarget filters," in *Signal and Data Processing of Small Targets 2000* (Drummond, O., ed.), vol. SPIE Proc. 4048, pp. 310–320, 2000.

[56] Mahler, R. P. S., "Multitarget Bayes filtering via first-order multitarget moments," *IEEE Trans. Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1152–1178, 2003.

[57] Mahler, R. P. S., "'Statistics 101' for multisensor, multitarget data fusion," *Aerospace and Electronic Systems Magazine, IEEE*, vol. 19, pp. 53–64, Jan 2004.

[58] Mahler, R., "Information for fusion management and performance estimation," in *Signal Processing, Sensor Fusion, and Target Recognition VII* (Kadar, I., ed.), vol. SPIE Proc. 3374, pp. 64–75, 1998.

[59] Menon, P. and Sharma, V., "Adaptive target state estimation using neural networks," in *Proc. of the American Control Conference*, vol. 4, (San Diego, CA), pp. 2610–2614, June 2-4 1999.

[60] Mori, S. and Chong, C. Y., "Point process formalism for multiple target tracking," in *Proc. FUSION 2002*, (Annapolis, Maryland), pp. 10–17, 2002.

[61] Orton, M. and Fitzgerald, W., "A Bayesian approach to tracking multiple targets using sensor arrays and particle filters," *IEEE Trans. on Signal Proc.*, vol. 50, no. 2, pp. 216–223, 2002.

[62] Panta, K., Vo, B.-N., Singh, S., and Doucet, A., "Probability hypothesis density filter versus multiple hypothesis tracking," in *Signal Processing, Sensor Fusion, and Target Recognition XIII* (Kadar, I., ed.), vol. Proc. SPIE 5429, pp. 284–295, April 2004.

[63] Papoulis, A. and Pillai, S. U., *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 4 ed., 2002.

[64] Pulford, G., "Taxonomy of multiple target tracking methods," *IEE Proc. Radar, Sonar and Navigation*, vol. 152, pp. 291–304, October 2005.

[65] Punithakumar, K., Kirubarajan, T., and Sinha, A., "A multiple-model probability hypothesis density filter for tracking maneuvering targets," in *Signal and data processing of small targets* (Drummond, O. E., ed.), vol. Proc. SPIE 5428, pp. 113–121, April 2004.

[66] Sastry, C. R. and Kamen, E. W., "SME filter approach to multiple target tracking with radar measurements," *IEE Proc. F: Radar, Sonar, and Navigation*, vol. 140, pp. 251–260, August 1993.

[67] Saulson, B. and Chang, K., "Comparison of nonlinear estimation for ballistic missile tracking," in *Signal Processing, Sensor Fusion, and Target Recognition XII* (Kadar, I., ed.), vol. Proc. SPIE 5096, pp. 13–24, 2003.

[68] Sidenbladh, H., "Multi-target particle filtering for the probability hypothesis density," in *Proc. FUSION 2003*, (Cairns, Australia), pp. 800–806, 2003.

[69] Sidenbladh, H. and Wirkander, S.-L., "Particle filtering for finite random sets." Downloaded May 2005 from http://www.nada.kth.se/∼hedvig/publications_en.html.

[70] Sidenbladh, H. and Wirkander, S.-L., "Tracking random sets of vehicles in terrain," in *IEEE Workshop on Multi-Object Tracking*, (Madison, Wisconsin), 2003.

[71] Sinha, A., Nandakumaran, N., Sutharsan, S., and Kirubarajan, T., "Tracking spawning targets with a tagged particle filter," in *Signal Processing, Sensor Fusion, and Target Recognition XIII* (Kadar, I., ed.), vol. Proc. SPIE 5429, pp. 13–22, 2004.

[72] Tobias, M. and Lanterman, A. D., "Multitarget tracking using multiple bistatic range measurements with probability hypothesis densities," in *Signal Processing, Sensor Fusion, and Target Recognition XIII* (Kadar, I., ed.), vol. Proc. SPIE 5429, pp. 296–305, April 2004.

[73] Tobias, M. and Lanterman, A. D., "A probability hypothesis density-based multitarget tracker using multiple bistatic range and velocity measurements," in *Proc. of the 36th Southeastern Symposium on System Theory*, (Atlanta, GA), pp. 205–209, March 2004.

[74] Tobias, M. and Lanterman, A. D., "Probability hypothesis density-based multitarget tracking with bistatic range and Doppler observations," *IEE Proc. Radar, Sonar and Navigation*, vol. 152, pp. 195–205, June 2005.

[75] Vo, B.-N. and Ma, W. K., "A closed form solution for the probability hypothesis density filter," in *Proc. FUSION 2005*, (Philadelphia, Pennsylvania), 2005.

[76] Vo, B.-N., Singh, S., and Doucet, A., "Sequential Monte Carlo implementation of the PHD filter for multi-target tracking," in *Proc. FUSION 2003*, (Cairns, Australia), pp. 792–799, 2003.

[77] Vo, B.-N., Singh, S., and Doucet, A., "Sequential Monte Carlo methods for multitarget filtering with random finite sets," *IEEE Trans. Aerospace and Electronic Systems*, vol. 41, pp. 1224–1245, October 2005.

[78] Wan, E. A. and van der Merwe, R., "The unscented kalman filter," in *Kalman Filtering and Neural Networks* (Haykin, S., ed.), ch. 7, John Wiley and Sons, 2001.

[79] Welch, G. and Bishop, G., "An introduction to the Kalman filter," Comp. Sci. Tech. Report 95-041, UNC-Chapel Hill, Apr. 2004.

[80] Willis, N., *Bistatic Radar*. Technology Service Corporation, 1995.

[81] Zajic, T., Hoffman, J., and Mahler, R., "Scientific performance metrics for data fusion: New results," in *Signal Processing, Sensor Fusion, and Target Recognition IX* (Kadar, I., ed.), vol. SPIE Proc. 4052, pp. 172–182, 2000.

[82] Zajic, T. and Mahler, R., "A particle-systems implementation of the PHD multitarget tracking filter," in *Signal Processing, Sensor Fusion, and Target Recognition XII* (Kadar, I., ed.), vol. Proc. SPIE 5096, pp. 291–299, 2003.

[83] Zajic, T., Ravichandra, B., Mahler, R., Mehra, R., and Noviskey, M., "Joint tracking and identification with robustness against unmodeled targets," in *Signal Processing, Sensor Fusion, and Target Recognition XII* (Kadar, I., ed.), vol. Proc. SPIE 5096, pp. 279–290, 2003.

# VITA

Martin Tobias was born in Cape Town, South Africa, in 1978, and grew up in Birmingham, Alabama. He received his A.B. degree *summa cum laude* from Harvard University in June 1999 and worked at start-ups IronBridge Networks and Sandburst Corporation before beginning graduate school at the Georgia Institute of Technology in the fall of 2002. He received the M.S. degree from Georgia Tech in December 2003 and plans to receive the Ph.D. degree under the supervision of Dr. Aaron Lanterman in May 2006.